OCCASION

This publication has been made available to the public on the occasion of the 50th anniversary of the United Nations Industrial Development Organisation.

DISCLAIMER

FAIR USE POLICY

CONTACT

Please contact publications@unido.org for further information concerning UNIDO publications.

For more information about UNIDO, please visit us at www.unido.org

UNITED NATIONS INDUSTRIAL DEVELOPMENT ORGANIZATION
Vienna International Centre, P.O. Box 300, 1400 Vienna, Austria
Tel: (+43-1) 26026-0 · www.unido.org · unido@unido.org

EMERGING TECHNOLOGY SERIES

Trends in Parallel Processing

Prepared for UNIDO
by
Boleslaw K. Szymanski (ed.), Lalit M. Patnaik and Sian Wun Song

UNITED NATIONS
INDUSTRIAL DEVELOPMENT
ORGANIZATION

EMERGING TECHNOLOGIES SERIES

Technology is now at the core of competitive strategies of successful industrial firms. The new and rapidly evolving generic technologies, such as biotechnology, new materials and information technologies, offer many opportunities and challenges for broad competitive strategies. They engender entirely new products, services, markets and businesses. Their impact is trans-sectoral, radically improving the competitiveness of products, processes and services of firms in a large number of traditional industrial sub-sectors. New materials improve product specifications and lower production costs in engineering and chemical industries; biotechnologies save energy and raw materials in chemicals, pharmaceuticals and food processing, while the pervasive applications of information technologies allow companies in all industrial sectors to re-engineer critical processes, improve overall efficiency and raise productivity across functional areas. Monitoring and access to information is now a key to competitiveness.

Experience in newly industrialized countries shows that access to reliable technical information can be instrumental in allowing manufactureres to leap whole periods of technological development and adopt state-of-the-art systems directly – without needing to undertake a painful and costly development phase. Up-to-date economic information and analysis of global economic trends and the prevailing industrial situation in other countries is likewise indispensable – and the gateway to identifying industrial needs, opportunities, constrains and priorities of the country and region concerned. Monitoring technological advances and economic analysis provide the basis for the formulation and effective implementation of appropriate industrial programmes and projects by both public and private entities. For developing countries, with their limited resources and often greater susceptibility to the negative aspects of technology-led change, such activities are doubly important. Yet many developing countries still lack the critical elements for technology monitoring of emerging technologies and their implications for national development strategies. If they are to maximise the benefits and minimize the negative effects of technology on social and economic development, developing countries must manage technology in an appropriate manner – and monitoring is an essential element of that management process.

One of the objectives of UNIDO is to carry out a set of coherent activities at the national, regional and international levels, to help developing countries at different stages of development to acquire, apply, develop and manage technologies against a global background of technological change. Investment and technology play a vital role in the industrial growth of developing countries, as well as their gradual integration into the international economy. Although most developing countries now have liberal regimes for investment and technology transfer, this is not a sufficient condition for industrial growth. There is a need for a wide-ranging investment and technology approach that will not only attract and retain the inflows of investment and technology, but also make the optimum use of them for the domestic economy. UNIDO's wealth of experience in industrialization, combined with its worldwide network of contacts makes the Organization an ideal partner to assist developing countries in building up their investment and technology partnerships. The Organization is a focal point of industrial technology; it is a global source of industrial information; and it is an honest broker for industrial cooperation.

Through this new series of publications on emerging technologies in developing countries, which supercedes the ***Industrial Technology Monitors*** and the ***Technology Trends Series***, UNIDO plans to sensitize industry and governments to the need for and requirements of technology monitoring and assessment in the areas of new and emerging technologies. These technologies play a catalytic role in the development process of the new global pattern of rapid and accelerating technological change, sweeping trade liberalization, far-reaching deregulation of markets – including the privatization of state-owned enterprises and commercialization of R&D – and the globalization of international business.

Investment and Technology Promotion Division
March 1996

AUTHORS

Boleslaw K. Szymanski (author and editor)
Professor of Computer Science
Department of Computer Science & Scientific Computation Research Center
Rensselaer Polytechnic Institute
Troy, NY 12180
USA
e-mail: szymanski@rpi.edu

Lalit M. Patnaik
Professor, Department of Computer Science & Automation
Professor, Supercomputer Education & Research Centre
Convenor, Microprocessor Applications Laboratory
Indian Institute of Science
Bangalore – 560012
India
e-mail: lalit@micro.iisc.ernet.in

Sian Wun Song
Associate Professor
Universidade de São Paulo
Instituto de Matemática e Estatística
Departamento de Ciência da Computação
CEP 05508-900 São Paulo, SP
Brazil
e-mail: song@ime.usp.br

CONTENTS

PREFACE

As part of UNIDO's programme of monitoring technological advances, the aim of which is to build up an awareness in developing countries of the emerging technologies and to help them strengthen their national technology capabilities, the *Emerging Technologies Series* has been initiated as a publication combining the well established *Industrial Technology Monitors* and the *Advanced Technology Trends Series*. Through this new series of awareness publications on technological advances, UNIDO plans to sensitize industry and governments to the need for and requirements of technology monitoring and assessment in the areas of new and emerging technologies. These technologies play a catalytic role in the development process of the new global pattern of rapid and accelerating technological change, sweeping trade liberalization, far-reaching deregulation of markets, including the privatization of state-owned enterprises and commercialization of R&D, and the gobalization of international business.

In continuation of our review of the latest developments, we consider that the recent developments in software engineering is an area that is of particular interest to developing countries—an area which they may not yet be fully aware of. For example, it has been forecast that the market for high-performance computing business will continue to grow to over US$4.5 billion in 1998 in the United States of America alone, and as widely reported, the trend by developed country software houses to subcontract software writing to small firms in Asia has been very successful and is growing. The explosive growth of the IT industry has allowed companies to create a competitive environment, and the outsourcing decisions made by a great many organizations show that about half of them outsource such services as data centre operations, telecommunications, applications development and applications support. The processes that are used to manage IT will determine how effectively a company controls the IT services that it consumes, and those companies that excel in developing such processes will end up not only with superior IT, but will have a superior ability to recognize and exploit market change.

One of the topics selected to be reviewed was parallel processing. Typical applications for massively parallel processors would be the monitoring of financial analyses, market analyses, process-intensive decision support and databases. Resulting from technological advances over the last few years, parallel processing is gaining momentum and with the constantly decreasing price of hardware, accompanied by the increase of processing power, this technology will soon become a major issue in information technology. In addition, the price of entry into parallel processing has decreased significantly during the past few years, creating many opportunities for new enterprises in the software industry in this area.

The timely presentation of a UNIDO publication on this subject would well serve decision makers in developing countries to assess and upgrade their own capabilities in this direction.

Konrad Fialkowski
Scientific Editor
Microelectronics Monitor

I. Parallel Processing – An Introduction

by

Boleslaw K. Szymanski

This document presents the current state-of-the art in parallel processing. In this section, we start with the overall introduction to the problems and challenges of parallel computing. The more detailed and regional perspectives are described in the sections that follow.

The Role of Computers

It is widely recognized that computer technology has become a critical component of everyday life in modern society. The computer has become ubiquitous in manufacturing, services, products and entertainment. Computers have been changing the ways in which we conduct business, produce goods and carry out science. Export controls introduced on certain computer equipment and the secrecy surrounding some computer projects clearly indicate this technology's importance in military and security services. An exponential growth in the power of computers, with computer centres being equipped with evermore powerful machines in the 1960's and 1970's, was followed by an exponential growth in the number of computers during the so-called personal computer (PC) revolution in the 1980's. We are currently undergoing yet another stage of the same process, an exponential growth in interconnectivity and bandwidth of the the network joining the computers together.

Computer literacy is becoming a norm, not the exception, and many knowledgeable workers, managers and other professionals now have the technical ability and skills to write software. According to [1], there are nearly two million people in the United States of America alone who work directly with software, and about ten million managers, engineers, architects, accountants and other knowledge workers who know enough about programming to be able to build end user applications with high-level tools (spreadsheets, databases, visual languages, etc.). Similar ratios are found in other industrialized countries around the world. On a global basis, there are more than ten million professional software personnel and more than 30 million end users who can programme. Table 1 gives rough estimates for each group in ten countries around the world (data are with a high margin of error).

The end-user programming population seems to be growing at more than ten per cent per year worldwide. The growth rate for software professionals is now down to a single digit in industrialized countries. In developing countries, the number of both end users and professional programmers is still growing at double digit rates. In view of these developments, it is

Table 1: Professional software personnel and end-users who program – 1995 estimates [1]

Country	Software Professional	End Users with Programming Skills	Percentage of workforce
USA	1,750,000	10,000,000	9
Japan	850,000	3,500,000	6
United Kingdom	385,000	1,750,000	6
France	375,000	1,700,000	7
Germany	350,000	1,650,000	4
Brazil	475,000	1,500,000	3
China	950,000	1,250,000	<1
India	750,000	1,200,000	<1
Russia	750,000	900,000	1
South Korea	300,000	750,000	5

not an exaggeration to compare the impact of computers on society to that of the Industrial Revolution in the 18th century.

The Industrial Revolution freed workers from the enslavement of manual labour and transformed crafts and handywork into the mass-producing industries of today. Likewise, the Computer Revolution we are now witnessing has been freeing office workers from routine mental tasks which were, and often still are being done by assistants, clerks and low-level managers.

Significance of Parallel Processing

Parallel processing is currently a small fraction of the overall computer technology and the Computer Revolution, yet there are two compelling reasons for parallel processing to be of much higher importance than indicated by its current share of computer technology. The first reason is that parallel processing is the basis of the most powerful computing engines, which are irreplaceable in the sciences, medicine and the drug industry. Large-scale computer modeling enabled by parallel processing impacts decision making in banking and finance, military and government. Parallel computers empower decision makers, such as high-level managers, military leaders and chief scientists, with the ability to gather, access, and synthesize information, as well as to simulate real-life processes to measure the impact of social, economic and design decisions. The quality of the simulations and synthesized

information is strongly dependent on the applied computational power. Today, even the largest uniprocessor computers are too slow for the most challenging problems of this kind.

The second reason for the importance of parallel processing is the dominant presence of sequential computing in the transient situation of today. There are clear indications, discussed below, that the ability of doubling the uniprocessor speed every 18 months, as has been done for the last decade, cannot continue and that the processor design technology is maturing. Interesting comparisons of this process to historic technological breakthroughs are presented in [2]. The authors contend that virtually every industry more than a few decades old has had to endure similar phase changes caused by the principle of economics and of supply and demand. The history of such old industries as aviation, automobiles and railroads could be used as pointers of what to expect of the semiconductor industry.

The first example the authors of [2] consider is aviation. Like the semiconductor industry, aviation went through a period of rapid growth. In less than four decades the industry moved from the Wright brothers' monoplane to the PanAm Clipper jet and the Super-fortress bomber. The initial growth of aviation was fueled by the military markets before moving on to civil transportation, again much like the semiconductor industry. Progress in aviation was made by increasing the speed of aeroplanes (thus reducing transit time) and by lowering the costs of moving a ton of cargo per mile travelled. Such dual progress is comparable to the computer processor's ever-increasing speed (thus boosting the power of the computer), while lowering the processor price. After several decades of growth in passenger capacity and airspeed, these trends peaked with Boeing's 747 as the highest mark for capacity, and the Concorde as the one for speed. Further progress was stopped by economic constraints, at least in civil aviation; for military applications cost is not a primary consideration (fighters produced by many manufactures exceed the speed of Concorde). For the 747, the difficulty was in filling the available space on all but the longest or most popular routes. For Concorde, the cost of fuel and noise pollution limited its usefulness as well. After these technological marvels, aviation entered a second phase in which a plethora of smaller, slower aeroplanes were designed and produced for more specific markets. The focus of research and development shifted from speed and size to more efficient and quieter operation and passenger comfort.

Another example given in [2] is railroads. The research there focused on increasing the power of locomotives to lower the cost of transportation. This trend peaked with the EMD DD-40, a monster locomotive that was too big and inflexible for any other purpose than hauling freight across the USA. These limitations resulted in the increased use of smaller engines that could operate separately for small loads, but could be joined together to transport large loads. The authors conclude that today, the semiconductor industry is in a situation similar to that of the railroad companies just before the EMD DD-40 was designed. The high cost of developing the factories for future generation chips forced the semiconductor companies to join forces in order to try and economically manufacture extremely dense chips.

The final example used in [2] is that of the automobile industry. Ford's initial success in car production resulted from lowering costs by concentrating production in ever-larger factories. This trend led to a diminished ability to vary products. In the early 1930's, General Motors recognized that large factories were only good for building large numbers of the same product, and that at the critical size already achieved, efficiency no longer increases along with factory size. Therefore General Motors split the company into divisions, with clearly defined markets and factories dedicated to support them. The resulting wider variation in designs allowed GM to gain a market share at Ford's expense.

A similar scenario is happening today in the semiconductor industry. Intel offers more than 30 variations of its 486 microprocessor, while in the early 1980's the company offered just three versions of its 8086 microprocessor. The authors conclude that in technology driven industries the initial phase is dominated by improvements both in performance and costs. The second, mature phase, is characterized by product refinement and diversity – similar to what is now beginning to happen in the semiconductor industry. Slowing the rate of progress in semiconductors will provide a more stable environment for computer architectures and software. As a result, parallel processing will became more widespread than it is today.

Applications of Parallel Processing

In the United States of America, the quest for faster machines has been fueled by computationally intensive problems with profound economic and social impacts, referred to as Grand Challenges [3]. It is difficult to list all Grand Challenge problems because so many areas of science and engineering are potential sources of such problems. A short list would typically include:

- High-resolution weather forecasting crucial for agriculture, disaster prevention, etc.
- Pollution studies that include cross-pollutant interactions, important in environmental protection.
- Global modelling of atmosphere-ocean-biosphere interactions to measure the long-term impact of human activities on the stability of the global ecosystem.
- Human genome sequencing that will assist in recognizing, preventing and fighting genetic diseases.
- The design of new and more efficient drugs to cure cancer, AIDS and other diseases.
- High-temperature superconductor design that can revolutionize computer design, electrical devices, etc.

- The aerodynamic design of aerospace vehicles (airflow modelling) and improvements in automotive engine design (ignition and combustion modelling) that can lead to a more efficient use of depletable fossil fuels in transportation.
- The design of quantum switching devices, important for the building of more powerful computers.

US research support agencies, such as the National Science Foundation, various agencies in the Department of Defence, the Department of Energy and the National Aeronautical and Space Agency, together fund research projects directed towards Grand Challenges. This is a five-year effort referred to as the High Performance Computation and Communication Program, or HPCC in short. It began in 1993 and has a yearly budget of several hundred million dollars. The investigations conducted under the HPCC Program involve multidisciplinary teams of researchers in the natural sciences, applied mathematics, and computational and computer science from different institutions. As an example, the author of this introduction is involved in four projects concerning the Grand Challenges problems being investigated at the Rensselaer Polytechnic Institute. Two of these projects are funded directly from the HPCC Program, with the two others being indirectly funded. One of the projects focuses on modelling human joints, while another focuses on shoulders and knees. The research is conducted in cooperation with the Orthopaedics Division of the Columbia University Medical Center. The goal is to be able to guide surgeons in operating on malfunctioning joints by simulating the behaviour of a joint under different operating scenarios. The joints are modelled by adaptive meshes, while finite element methods are used to solve the partial differential equations describing the joints' behaviour. Another project focuses on problem solving environments for the optimization and control of chemical and biological processes. This investigation is conducted in cooperation with groups at the University of Minnesota and the University of California at San Diego. The primary goal of this research is the development of a high-performance problem solving environment (PSE) for the optimization and control of chemical and biological processes, with an initial emphasis on bioengineering applications. The optimization and control of such processes requires the repetitive solution of time-dependent partial differential equations (PDEs) in two or three spatial dimensions. The computational requirements of this problem, which must be solved interactively, can only be met by useing massively parallel computers. Such a comprehensive and powerful PSE does not currently exist, and its development presents significant computational and computer science challenges. The third project is part of a tokamak design, the ultimate goal of which is to build the sustainable plasma generation device supported by hot fusion. The purpose of our investigation is to develop a scalable and portable Plasma in Cell (PIC) for the simulation of plasma behaviour in a self-generated electromagnetic field. This work is being done in cooperation with other researchers at the University of California at San Diego and the Jet Propulsion Laboratory. Finally, the fourth project focuses on individually based modelling of epidemics. In cooperation with biologists from the State University of New York in Albany we are investigating the spread of Lyme disease and the ways of controlling its proliferation. Our

basic computational tool is a 36-node IBM SP2 parallel computer with a peak performance of about 9 gigaflops (i.e., 9 billion floating point operations per second) available on the campus. This machine is used mainly for code development and test runs. The production runs are conducted on a 400-node SP2 at the Maui (Hawaii) Supercomputing Center and a 512-node SP2 at the Cornell Supercomputing Center located at Cornell University. Both of these machines have a peak performance to the order of a hundred gigaflops. The research on plasma simulation involves additional machines, a Cray T3D and an Intel Paragon at the Jet Propulsion Laboratory, as well as a network of Sun workstations at the Rensselaer Polytechnic Institute. This example of the research involvement of a single scientist perhaps best describes how diversified the HPCC Program is, and how much cooperation it has fostered.

Required Computational Power

It is estimated that to achieve interactive response time for Grand Challenge problems, in the order of minutes for smaller instances and hours for larger ones, will require a machine with a performance of teraflops (which is a thousand billions of floating point operations per second). Several architectures today have a theoretical peak of teraflops, with the cost below US$ 100 million (e.g., the AVALON computer based on the DEC Alpha chip and fast interconnection). However, sustained performance has been demonstrated at the level of tenths of teraflops, i.e., about a few 100's gigaflops. Even in those cases, such speed was achieved only on certain very large, highly localized, finely tuned, and often idealized applications. The real drawback is in the software and the ability to find enough useful parallelism in an application to use all the computer power efficiently. Yet parallel processing is the only viable option for sustained growth in computer performance, in view of the imminent stalemate in the semiconductor industry discussed earlier. In addition to economical forces (exponentially increasing costs of hardware needed to fabricate chips with smaller dimensions), there are basic laws of physics that put a limit on the speed of a uniprocessor. The speed of signal transmission in a computer cannot exceed the speed of light in the transmission media, which is about 300,000 km/sec. for silicon. Consequently, it takes one billionth of a second for a signal to propagate on a silicon chip an inch in diameter. However, one signal propagation can support at most one floating point operation. Hence, a sequential computer built with a chip of such a size can provide at the most a gigaflop of performance, which is merely one-thousandth of the needed teraflops.

Parallel Architectures

The interest in parallel computing systems is not new and can be traced back as far as the 1920s. However, as late as the early 1970s, major criticism of parallel processing was based on Grosch's law, which states that the computing power of a single processor increases in proportion to the square of its cost. Recent careful analysis of Grosch's law showed that it is valid only within one technology. Economy of scale for mass-produced memory and RISC (Reduced Instruction Set) processors makes them a few orders of magnitude less expensive

than custom designed chips for mainframes and traditional vector supercomputers. The improving computer chip technology enables the placement of ever faster processors with ever increasing amounts of memory on a single wafer. Hence, the introduction of RISC technology made Grosch's law obsolete. Massively parallel computers built from a large number of RISC processors provide a superior performance-to-price ratio compared to computers based on the powerful, custom-designed CISC (Complex Instruction Set) processors.

The traditional vector supercomputers are built of a limited number of powerful, specially designed processors connected to a large shared memory. In addition, they explore array operation parallelism through vector co-processors. However, the support for shared memory limits the number of processors that can be clustered together in such a way that all have the same access time to the whole memory. Hence, purely shared memory machines are not scalable. In contrast, massively parallel computers consist of off-the-shelf processors with local memories. The processors are connected directly to each other by a network. The cost of such a parallel computer is roughly proportional to the needed number of processors. Therefore the size of the computer installation is more limited by costs than technical considerations. In addition, the market forces of general computing, which are two orders of magnitude larger than the parallel computing market of today, drive the process of technological progress for processor design, resulting in a fast increase in processor speed and a decrease in processor price. Hence, the massively parallel computers have three advantages over traditional vector supercomputers:

1. An accelerated rate of advance of peak processing power. In the last decade, microprocessor performance has increased four times every three years, following the rate of integrated circuit logic density improvement. By contrast, the clock rates of vector machines have improved much more slowly, doubling every seven years [3]. These trends are expected to continue for at least the rest of the 1990s.

2. An improvement in the performance-to-cost ratio. In 1993 this ratio was between two to eight times higher for MPPs than for the vector supercomputers.

3. Scalability of the machine. The smallest configurations of MPPs are usually low priced to entice initial purchase (in 1995, the least expensive MPPs cost below $50,000). The initial configuration of the MPP can be upgraded incrementally as needs arise and funds become available.

The clear conclusion is that only massively parallel computers can deliver the much needed teraflops level of performance. Parallel programming has experienced a long and difficult maturation process. The reasons are many, but perhaps the most critical one is the difficulty in programming the newly developed architectures. Porting and tuning an application to a new architecture can take as long as the interval between the introduction of new architectures, making a newly developed code obsolete at the moment of its creation. In such an environment, programmers face a daunting challenge, especially with increasingly large

and complex applications. Programmers must identify parallelism in an application, translate that parallelism into code, and design communication and synchronization for the programme, all in the context of currently available architectures, which may change tomorrow, making some of the designs suboptimal or inefficient.

Parallel Programming

One of the promising approaches to curb the cost of parallel software redevelopment is object-oriented programming. However, according to Grimshaw [4], the object-oriented parallel programming community is divided over the issue of how to support parallelism in an application. There are two primary schools of thought. The first, the libraries group, argues for building highly optimized, extensible class libraries that encapsulate parallelism. Users could use these class libraries without knowing anything about parallelism or about what goes on inside the class library. The heart of the library group's argument is that C++ already provides a powerful mechanism for language extension, viz., classes, inheritance and templates. Additional extensions would only clutter the language. Furthermore, with no consensus on language features, compiler vendors are unlikely to support any language extensions, and users will not want to risk embracing the "wrong" feature.

The second school of thought, the extension group, argues that the best way to achieve parallelism is via language extensions. The heart of the argument is that parallel composition is as important a concept as sequential composition. With concurrency being a part of the language, compiler technology can more readily develop parallel code optimizations.

Grimshaw [4] believes that parallel processing is at a crossroad. In the past, parallel processing mainly relied on expensive supercomputers with software often being developed in-house, because the commercial software developers considered this market as being too narrow to be financially viable. Instead, a booming commercial software business targets personal computers and work stations. Today, however, these cost-effective desktop computers have closed the performance gap. At the same time, many traditional parallel processing users are downsizing and no longer have the resources to develop everything in-house. Therefore, the parallel processing community has tremendous incentive to leverage the existing commercial software base.

Leveraging commercial software is crucial. The scientific/parallel software market is miniscule compared to the desktop computer market. The scientific computing community does not have the resources to duplicate software development efforts. Therefore it must adapt, conform to existing standards and exploit the desktop market, a market that is increasingly moving towards object-based and object-oriented interoperability standards.

It is difficult to use parallel components developed by different research groups in a single application. There is also a desire to construct multidisciplinary simulations, for example coupling ocean and atmospheric models in a global climate model. The individual components of these simulations are often stand-alone parallel codes. While the system file can be

used as an interface and data transport mechanism, more efficient techniques are needed. Object technology can be used as an interface description mechanism, a data transport and coercion mechanism and as a mechanism to extend the life of the legacy components.

If parallel processing components conformed to standard interface descriptions, they could be used transparently by commercial application developers. That would make parallel computing relevant to a broader user base and encourage vendors to develop parallel hardware for the commercial markets. The feasible way of doing this is to encapsulate parallelism within objects, making the parallel component a particularly fast version of an existing sequential code.

From that perspective, the decision of the High Performance Fortran (HPF) designers to base the language on Fortran90 was very helpful. Fortran90 includes all the basic constructs required for object oriented programming and there is an increasing interest in object-oriented programming using Fortran90.

Summary of the following sections

The above trends focus on the global perspective of parallel programming. The rest of this document provides more detailed and regional points of views on these issues.

The USA Perspectives

First, in the section entitled "Trends in Software Engineering for Parallel Processing" the author assesses the current state-of-the-art in this area from the USA perspective. With about 50 per cent of the parallel computing power installed in the USA, national research and development programmes in the area of high performance computers have the highest number of personal computers and computer users. The US perspective is important for other countries as an indication of future developments and as a way to avoid deadends. The section starts with a discussion of the trends in architectural design. The US high performance computing industry has undergone a profound transformation. The emphasis shifted from record-breaking performance at any price, to price performance optimization. Successful companies, such as Silicon Graphics or IBM, use the performance gains driven by the general market to improve the performance of their parallel machines. In contrast, companies that relied on processors designed specifically for their architectures, like Kendall Square Research or Thinking Machine Corporation, were less successful in staying in the computer design market.

Another trend discussed in this section is the increasing importance of memory for the speed and economy of parallel processors. The increasing speed of the processors is matched by the increasing capacity, but not speed, of the memory. As a result, the gap between the speed of processors and memory widens every year. The gap is masked by the use of ever increasing cache. As evidenced by the pricing and performance of Silicon Graphics Challenger, the ratio of cache to memory improves the price performance of parallel machines on memory intensive applications.

The section also discusses trends in speed, price performance and distribution of parallel processors. Gordon Bell Awards awarded yearly in the USA indicate steady exponential growth in speed and price performance for the last decade. The parallel computing speed on useful applications reached several hundred gigaflops last year. The price-performance ratio is in the region of ten gigaflops per million dollars of hardware cost. The largest worldwide supercomputing sites are still dominated by the governmental centres. However, the medium sites are mainly industrial, while the smallest are mainly academic. This distribution contrasts with the overwhelmingly governmental centres in all categories just five years ago. This change in distribution clearly indicates that the impact of parallel processing on industries is growing.

Another topic discussed in this section is the development of software models for parallel processing. First, traditional models are discussed, such as Single Instruction Multiple Data (SIMD) and Single Programme Multiple Data (SPMD), which is a restricted version of the more general Multiple Instruction Multiple Data (MIMD) model. Then a new Bulk Synchronous Parallelism Model (BSP)is described, together with its library. The final pages of the section are devoted to trends in languages. In particular, the basic ideas behind High Performance Fortran (HPF) are described and compared to the Message Passing Interface (MPI) based approach.

Asian Perspectives

The third section of this document, entitled "High Performance Computing in India and the Far East" authored by Professor Lalit Patnaik from India, focuses on the Asian perspective of parallel processing. The author describes India's national initiative in supercomputing and its goal to develop a modern distributed memory parallel computer. The result was the PARAM-8000 parallel computer based initially on the INMOS transputer (1990-92). The sustained performance of the 16-node PARAM 8600 was in the range of 0.1–0.2 gigaflops. The next generation of PARAM computers, PARAM-9000, was based on the SuperSparc series processor, thus following the modern trend of using off-the-shelf mainstream processors in the parallel machine architecture. After discussing architectural details of these machines, Professor Patnaik describes its software environment, which includes support for HPF for data parallelism and PVM and MPI for MIMD parallelism. Then the author focuses on applications on PARAM, which currently are of a typically scientific computation mix, however the are plans to incorporate production quality industrial codes in the near future.

In the following pages Professor Patnaik describes several other Indian parallel computers developed by different governmental research centres. Discussing the performance capabilities of these machines, Professor Patnaik demonstrates that by employing a sufficient number of nodes, the PARAM-9000 can achieve a peak performance of teraflops. The support for Indian High Performance Computing activities comes from different agencies of the government.

In his section, Professor Patnaik also briefly summarizes the High Performance Computing activities in other countries from the Far East region. His summary discusses developments and programmes in Australia, China, Hong Kong, New Zealand, the Republic of Korea and Singapore. The section concludes with a brief assessment of the major trends in parallel computing.

Latin America Perspective

The penultimate section of this document, entitled "Parallel Computing: a Latin American Perspective" written by Professor S. W. Song from Brazil, concentrates on the Latin America region. First, the author identifies the major supercomputing sites in Latin America; two of them are located in Brazil and the other two in Mexico. Then, Professor Song describes the national supercomputing centrerrs in Brazil, which are supported through the Ministry of Science and Technology.

In his section, Professor Song describes computing in Latin America and compares the values of computer related products and services in several countries of the world with those of Brazil. To increase the production of high quality software, the Brazilian government and private industry launched a joint initiative called SOFTEX 2000. Next, Professor Song describes research in parallel computing in Brazil and Chile and existing international cooperation. There are links with research universities in the USA as well as with the European Union's ESPRIT programme. The section also contains a brief description of two interesting research programmes in Brazil, one focusing on methodologies of design for scalable algorithms, and the other on design of virtual shared memory systems. The section lists the supercomputing sites in Latin America and concludes with Professor Song's remarks on the role of governments and funding in future developments of parallel processing in this region.

Acknowledgement

As an author of sections of this publication and coordinating editor of the entire document, I would like to thank my co-authors, Professor Patnaik and Professor Song for their kind cooperation and valuable contribution to this publication. On behalf of all authors, I would also like to express our gratitude to Dr. K. Fialkowski of UNIDO for inviting us to write this document.

References

[1] Capers Jones, *End-user Programming*, **IEEE Computer**, Vol. 28, No. 9, September 1995, pp. 68-69.

[2] G. Dan Hutcheson and Jerry D. Hutcheson, Technology and Economics in the Semiconductor Industry, **Scientific American**, Vol. 274, No. 1, January 1996, pp. 54-63.

[3] L. Branscomb (Chair), *From Desktop to Teraflop: Exploiting the U.S. Lead in High Performance Computing,* NSF Blue Ribbon Panel on High Performance Computing, Washington, DC, August 1993.

[4] Andrew Grimshaw, *Parallel Processing and OT*, **IEEE Computer**, Vol. 28, No. 10, October 1995, pp. 63-64.

II. Trends in Software Engineering for Parallel Processing

by

Boleslaw K. Szymanski

Introduction

The increasing importance of parallel processing caused by its rapid growth encouraged the development of standards in parallel programming languages and tools, yet there is no evidence of a convergence of the supported paradigms to a single model. In this section, we review two of the current most popular models for parallel programme design: data parallelism and message passing. We also discuss the relevant developments in object oriented programming techniques, as well as in client-server distributed/parallel processing. The declining share of the parallel processing market held by traditional supercomputers and the waning popularity of SIMD machines, together with the increasing role of clusters of workstations, has created the right conditions for a rapid spread of parallel machines in government and industry. The price of entry into parallel processing decreased significantly, making abundant opportunities for new enterprises in software industry in this area. This section reviews these developments and also discusses changes in supporting areas of software engineering for high performance distributed computing. Finally, the section reviews the perspectives and impact of the changing parallel computing industry on information processing at international and national levels.

Trends in Architectural Design

Recent events, such as the filing for Chapter 11 bankruptcy by the Thinking Machine Corporation in August 1994, the discontinuation of manufacturing and sale of KSR supercomputers by Kendall Square Research in September 1994, and the disappearance of Cray Computer in 1995, raises the question of how secure is actually the future of the parallel computing industry.

It is important to realize that at present, parallel computing constitutes a small fraction of the overall information technology industry. In 1994, overall US information technology products were worth about $500 billion [5]. This value is a middle point of two estimates. The first estimate was provided by the US Department of Commerce and was based on data from the US Bureau of the Census. It values shipments for the information technology industry at $421 billion for 1993. This number includes computers, storage related devices, terminals and peripherals, packaged software, computer software manufacturing, data

processing, information services, facilities managements and other services, as well as telecommunication equipment and services. However, this number does not include revenue from equipment rentals, fees for after sale service and mark-ups in the product distribution channel, as well as office equipment.

The Computer and Business Equipment Manufacturers Association (CBEMA) values the worldwide 1993 revenue of the US information technology industry at US$ 602 billion. This number includes sales of office equipment, and CBEMA reports larger revenues for information technology hardware and telecommunications equipment than the reports provided by the US Department of Commerce.

In the United States of America, the total revenues of parallel computer hardware manufacturers were estimated at about US$ 1 billion in 1993. Out of this sum, about US$ 400 million was received by manufacturers of massively parallel machines. Even with services and software revenues, the parallel computer industry was about 0.5 per cent of the US information technology industry. Such a small percentage of the overall market indicates a narrow user base that can be easily saturated with new products. In addition, parallel computing has been highly dependent on government policies. Institutions and government supported universities traditionally constituted more than 50 per cent of all users.

The end of the Cold War and the associated shift of governmental spending in the USA drastically changed the market for parallel machines and supercomputers. As a result, companies relying solely on the manufacturing of parallel machines have suffered the most. At the same time, companies for which parallel computing manufacturing is only a part of the product line (e.g., IBM Corp., Silicon Graphics Inc., Intel Supercomputing) have persevered, and others (most notably Hitachi and NEC in Japan) have entered or expanded their presence in the parallel systems market.

Predictions about a lasting impact of the current changes on the parallel computing industry vary widely. Some see the beginning of the end of parallel computing based on massive parallelism in the recent bankruptcy protection requests. Others argue that it is just an end of a beginning. In the first camp is Gordon Bell, the founder of several computer companies and the sponsor of the yearly Gordon Bell Awards for the fastest parallel computer [1]. Mr. Bell believes that the latest threat to the very existence of the industry comes from standard workstations and fast, low-latency networks based on ATM. These networks, according to Bell, like massively parallel machines, offer size scalability (smooth transition from fewer to more processors). However, unlike parallel machines, they also support generational scalability (from previous to future hardware generations) and space scalability (from multiple nodes in a box, to computers in multiple rooms, to geographical regions). The most important capability offered by these networks is application compatibility with workstations and multiprocessor servers. This is a capability that massively parallel computers sorely lack. According to Bell, the weaknesses of massively parallel machines stem from the following two factors:

1. Parallel architectures are best suited to highly tuned, course-grained, and/or dataparallel problems;

2. Every new generation of parallel architectures differs from the previous one, forcingthe users to redevelop their applications.

Bell sees the future of parallel processing in networked workstations and shared-memory multiprocessors.

In a rebuttal to Bell's criticism, James Cownie from Meiko [4] cited the reasons why networked workstations are not an answer in many environments. The most important reason is the need for data security and availability. For a large commercial organization, security of data and its accessibility to those who need it are crucial. The solution is a single, central repository. However, the central repository could use the same components as workstations to amortize the cost of their development. The natural solution is to use multiple processors compatible with the workstations. High performance requires a small physical size because the speed of light limits the performance of highly distributed machines (to keep the latency of communication below one microsecond, the distance between computers must be kept below 300m). Switching technology cannot be based on ATM's in such a repository, because ATM switches are an order of magnitude slower and more costly than proprietary switching technology. According to Cownie, the only alternative is a massively parallel machine.

A similar point is raised by Philip Carnelley and William Cappelli of Ovum Ltd. [2]. They underline that effective manipulation of large amounts of data is crucial for companies in maintaining a competitive advantage in the market. The complex applications in manufacturing, commerce, travel and entertainment require a sophisticated database support that demands enormous computing power. The costs of hardware and application development restricted parallel processing to niche applications, such as scientific computing, weather forecasting, etc. Yet only parallel computing can meet the current challenge of information processing and, in response to those needs, parallel processing has entered the commercial mainstream. Parallel computers built from standard components (e.g., shared-memory, like Sequent, or distributed memory, like IBM SP2) can run powerful parallel relational databases. Such systems can process data extremely quickly, are reasonably priced, and are impressively scalable. Today, most of the commercial uses focus on data repository. Carnelley and Cappelli predict that future applications of parallel systems will transform operational systems, decision support systems and multimedia applications, and in the process, will provide an enormous impetus for the parallel computing industry.

Ken Kennedy, from Rice University [8], underlines that part of the difficulty in making parallel computing widespread and popular was the lack of standards in parallel programming interfaces. As discussed later in this chapter, such standards have been developed and are gaining widespread acceptance.

The optimistic views on the future of parallel processing are supported by an exponential growth in the use of parallel supercomputers at the NSF Supercomputing Centers in the USA (see Table 1).

Table 1: Supercomputing Usage at NSF Centers in the USA

Fiscal Year	Active Users	Usage in Normalized CPU Hours
1986	1,358	29,485
1987	3,326	95,752
1988	5,069	121,615
1989	5,975	165,950
1990	7,364	250,628
1991	7,887	361,037
1992	8,758	398,932
1993	7,730	910,088
1994	7,431	2,249,562

(Source: National Research Council [5]

Some analysts see the exponential growth of revenues for massively parallel computers in the near future. Terry Bennet, director of technical systems research for Infor-Corp. in Beaverton, Oregon, was quoted in [12] as saying that the industry is currently in a "lag" where traditional vector supercomputers are fading out while other approaches are maturing. Bennet predicts that by 1996 there should be a reasonable upswing in the high-performance computing business and the market will continue to grow over US$ 4.5 billion in 1998. The strong sales of relative newcomers to the market, IBM Corp. with its SP series and Silicon Graphics Inc. with the Challenger computer, agree with Bennet's prediction.

Steve Wallach of Convex [13] argues that parallel processing has been becoming ubiquitous on all levels of computing technology. In microprocessor design, super-scalar techniques – executing multiple instructions at the same time – are now a standard. Multiprocessor file servers are in the process of becoming a standard. The continuing increase in the semiconductor density (see Table 2) will naturally lead to multiple processors on one semiconductor die. If a standard 64-bit RISC microprocessor has 1-2 million transistors (without cache), what else (other than creating a multiprocessor chip) can be done with transistors when 100 million and one billion transistors become available? Wide-spread use will drive the costs of such a chip down and will therefore make massively parallel computing cost effective.

Table 2: Semiconductor Technology Trends

	1992	1995	1998	2001	2004	2007
Feature size (micron)	.5	.35	.25	.18	.12	.1
Gates per chip	300K	800K	2M	5M	10M	20M
Bits per chip in DRAM	16M	64M	256M	1G	4G	16G
Microprocessor chip size in square mm	250	400	600	800	2000	1250
Memory (DRAM) chip size in square mm	132	200	320	500	700	1000
Wafer diameter in mm	200	200	200-400	200-400	200-400	200-400

(Source: Semiconductor Industry Association, March 1993)

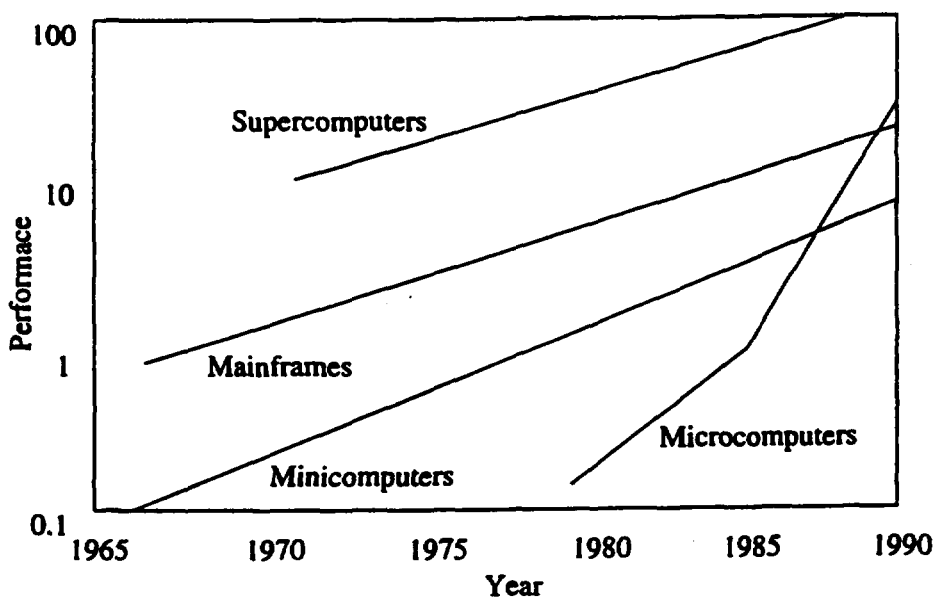


Figure 1: Trends in microprocessors and Mainframe CPU Performance Growth

The case against supercomputing and massively parallel computers is often based on the difference in speed with which the performance of microprocessors and other CPU's grew (see Fig. 1, which was based on [7]). However, the CPU performance gains are of one of two kinds:

1. Architectural advances: bit-parallel memory and arithmetic, cache, interleaved memory, instruction lookahead, instruction pipelining, multiple functional units, pipelined functional units and data pipelining;
2. Pure hardware advances, basically improvement in instruction cycle time, which is costly and limited by the physics of propagating signals through a medium and dissipating heat generated by transistor operation.

Microprocessors only relatively recently started to use architectural advances, whereas supercomputer CPU's used some of them before 1970s. Consequently, performance improvement resulting from some of the architectural advances is not seen in the plot for supercomputer CPU improvements. However, the pure hardware advances are based on advances in technology, which are increasingly costly. For example, the capital cost of a semiconductor fabrication line is growing rapidly with improvements in wafer and features sizes (see Fig. 2).

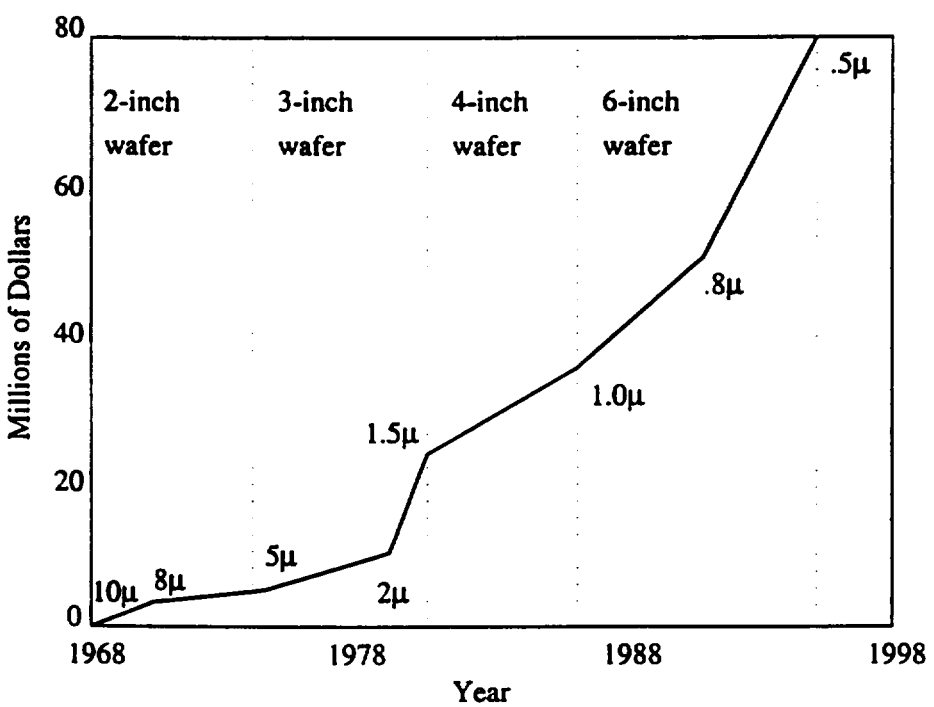


Figure 2: Semiconductor fabrication line capital cost per thousand wafers per week. Feature size is measured in microns. Source: [7].

This capital cost can be much more easily amortized if the produced chips are used not only in supercomputers or parallel computers, but also in all other lines of computers. Another

reason for using stock hardware in building parallel machines is the constantly improving performance of microprocessors. As shown in Fig. 3, designing a specialized processor for parallel processing, which has a tenfold performance advantage over the current uniprocessor design, gives the designers just four years of speed superiority. After that time, the improvements in general microprocessor design will nullify any initial performance advantage. Perhaps this is the reason why all three companies mentioned at the beginning of this section as getting out of the parallel processing manufacturing were using custom design chips in their products. On the other hand, the recently most successful parallel hardware manufacturers (IBM Corp. and Silicon Graphics, Inc.) use the standard CPU chips designed for their main line of workstations.

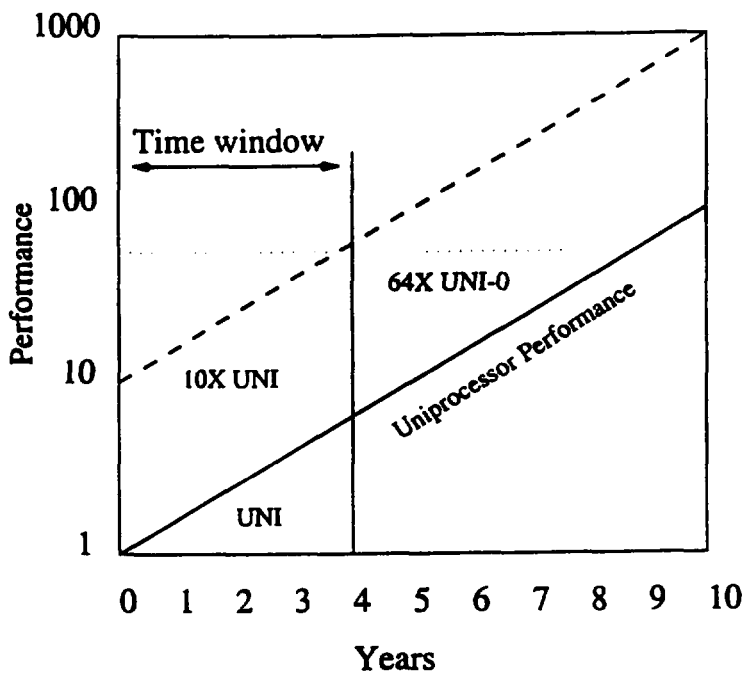


Figure 3. Performance window of opportunity for custom design chips

Importance of Memory

Technology is behind yet another phenomenon important for design and programming of parallel machines. There is a clear trend of DRAM speed improvements lagging behind the processor speed improvements (see Fig. 4). In the last twelve years, the CPU speed increased several hundred times, whereas the speed of DRAM chips merely doubled. Both chips are produced by the same technology, however the advancement in technology for DRAM chips is used to increase RAM density, not speed.

To mask the difference in speed between the processor and memory, modern processors use caching systems, often two level caches. A cache trades capacity for speed. During program

execution, the most recently referenced fragment of the memory is kept in the cache and data are retrieved from it. Each time data needed by the processor are already in cache, the access is done at (roughly) processor speed. Such an access is called a cache hit. When the data are not available in cache, cache miss happens, and a bucket of data (equal in size to the cache line) that contains the needed data is moved from a slow memory to cache. The access to data is slow in such a case. The cache miss ratio (or in other words the percentage of cache misses over all data accesses) dictates the resultant speed of processing. The bigger the difference in speed between the memory and the processor, the lower the cache-miss ratio must be for the processor to work at near capacity (see Fig. 5).

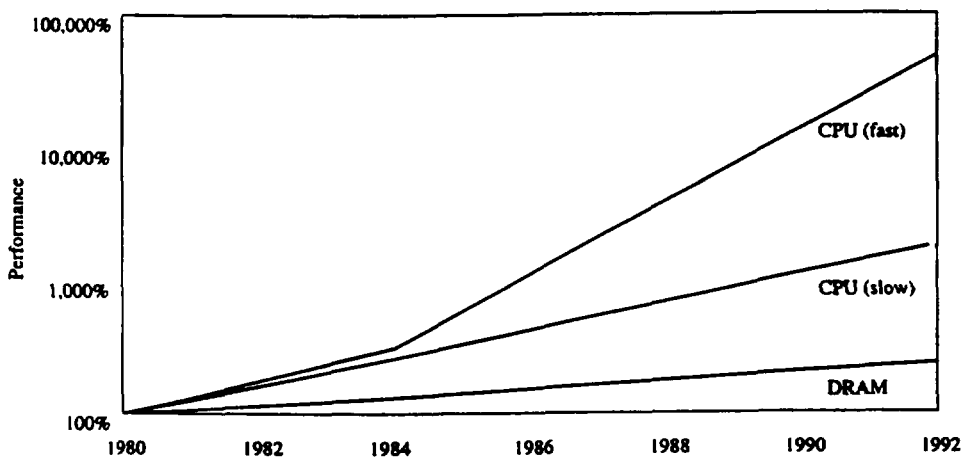


Figure 4: Trends in DRAM and processor cycle time: Source [7]

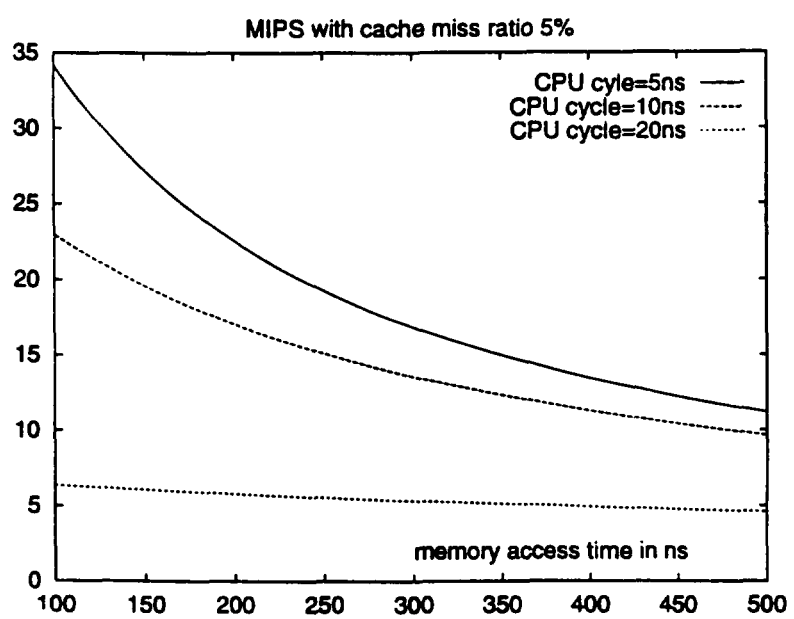


Figure 5: Effects of Memory-Access Time on Speed of Processing

In general, the cache miss ratio can be improved only by increasing the size of cache. Consequently, an increase in difference between the performance of the processor and memory is compensated by an increase in the cache size.

A multiprocessor brings not only multiple power of the CPU's together, but also improves memory capacity and performance, thanks to multiple cache capacity. As a result, a multiprocessor consisting of n processors may perform faster than n times the speed of a single processor for some applications. In such cases, the multiprocessor achieves super-linear speedup. It should be noted that the improved cache miss ratios of component processors of a multiprocessor must provide performance improvement that exceeds the overhead of parallel execution (such as load imbalance, not all processors having the same amount of work, and communication overhead, delayed access to non-local data), so cases of super-linear speedup are rare. However, in all applications, the impact of the extended memory of the parallel computer versus its single processor counterpart can be significant.

David Wood and Mark Hill discuss in [15] the concept of a costup and show that large memories can make parallel computing cost-effective even with modest speedups. Let $s(p)$ denote the speedup of a program when executed on p-processors, i.e.,

$$s(p) = \frac{V_{time(p)}}{V_{time(1)}} = \frac{time(1)}{time(p)}$$

The speedup is linear when $s(p) = p$, super-linear when $s(p) > p$, and sublinear (the most often case) when $s(p) < p$. Let $c(p)$ denote the cost of a p-processor machine. The cost-performance of such a machine, *costper f(p)* is then $c(p) * time(p)$. If a parallel machine is to achieve better cost-performance than a uniprocessor, then *costper f(p)* < *costper f(1)*, which leads to the following conclusion (see [15]):

> *p-processor parallel computing is more cost-effective than uniprocessor computing whenever* $s(p) > c(p)$.

The main point is that often $c(p) < p$ because processors in the multiprocessor may have less memory each than the uniprocessor. The authors provide an example of SGI systems. As of July 1994, a uniprocessor Challenge DM costed:

$$cost(1,m) = \$38{,}400 + \$100 * m$$

where m is memory size measured in Mbytes. The comparable p-processor, SGI Challenge XL, costed:

$$cost(p,a,m) = \$81{,}600 + \$20{,}000 * p + \$100 * a * m$$

where $a > 1$ is the factor of the memory overlap on different processors. By substitution, David Wood and Mark Hill obtained the following formula for SGI machines:

$$c(p,a,m) = (2.125 + 0.521 * p + 0.0026 * a * m)/(1 + 0.0026 * m)$$

Fig. 6 illustrates costups with SGI prices under the assumption that $a = 2$, i.e. that the parallel implementation requires twice the memory of the uniprocessor program. Different lines correspond to different numbers of processors p. The data support the assertion that parallel computing can be cost effective at speedups much less than p for large but practical memory sizes. Wood and Hill conclude that more than one processor may be needed to effectively utilize sufficiently large memories.

In the closing of this section, it should be noted that several different architectural approaches to parallel processing are slowly converging to a similar solution. The workstations interconnected through a fast network, when dedicated to a single application behave like a multiprocessor. The modern shared memory multiprocessor relies on an interconnection network between the global memory and local processor caches, and therefore behaves similarly to the distributed memory multiprocessor. Finally, distributed memory machines, through extensive use of caches, approach shared memory machines in their behaviour. The overall trend is to use powerful computing nodes interconnected through a high speed network of large capacity. The trend is to rely on standard, off-the-shelf components.

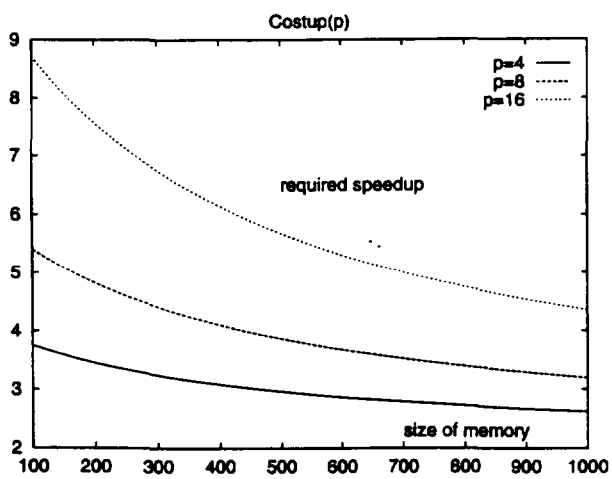


Figure 6. SGI costups with double memory overhead for a = 2.

Performance of Parallel Computers

As already discussed, the speedup *s(p)* of a parallel machine with p-processors can be found by comparing time of execution of a program by a uniprocessor *(time(1))* and by a multiprocessor *(time(p))*

$$s(p) = \frac{time(1)}{time(p)}$$

The well known adage that the chain breaks at the weakest link has a computer science counterpart in Amdahl's Law, which states that the least parallelizable part of the code limits the speedup. More precisely, if f is the fraction of the code, which is inherently

sequential (so-called Amdahl fraction), then independently of the number of processors used $s(p) \leq 1/f$, *simply because* $f * time(1) \leq time(p)$.

Amdahl's Law seems very pessimistic: after all, every program has sequential parts and even if these parts are small and limited to a few per cent of the code, still the speedup is limited to less than a hundred times (see Fig. 7).

Fortunately, the execution time of sequential parts of the algorithm do not often change, or change slowly with the growth of the problem size, whereas execution time of parallelizable parts changes rapidly when the problem size is increased. Hence, the Amdahl fraction is dependent on the problem size. For a wide class of problems f can be made arbitrarily small by selecting a sufficiently large problem size. Consequently, for such problems, the speedup can be made arbitrarily large.

Often the problems computed on parallel machines are too large to fit on a uniprocessor, so measuring an Amdahl fraction for them is impossible, or difficult. John Gustafson [6] proposed a different measure, g, that represents a fraction of time during which the parallel machine executed the sequential part of the code. Therefore $time(p) = g+(1-g) = 1$ but $time(1) = g+(1-g)*p = p-(p-1)*g$, so the speedup is:

$$s(p) = p[1 - (1 - 1/p) * g]$$

The nice feature of this formula is that it clearly shows how to improve the speedup. If we start adding processors (i.e., increasing p), but keep the work of all processors the same, then most likely g will stay the same and the speedup will grow. Likewise, with the constant number of processors, we decrease g by increasing the problem size. The final conclusion is similar to what Amdahl's Law implies: by selecting a large enough problem to keep all processors occupied for a long time, the impact of the sequential parts of the program could be made negligible.

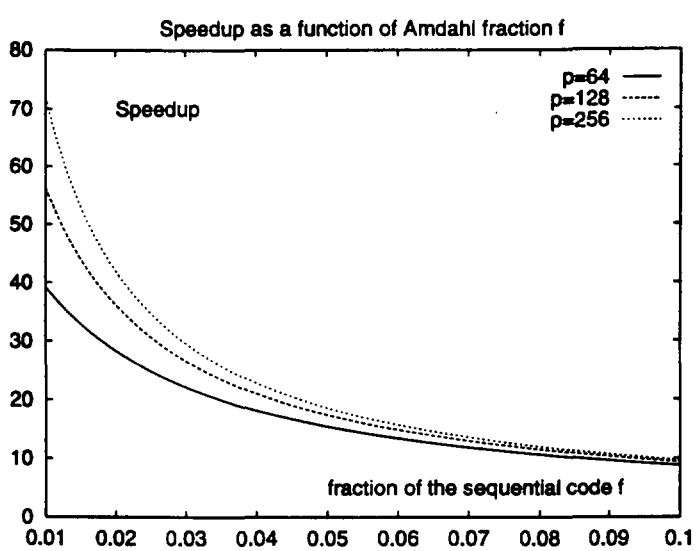


Figure 7. Impact of Amdahl's Law on the Maximum Speedup.

Although these principles may seem simple in theory, applying them to real problems is difficult. To encourage innovation, the annual Gordon Bell Awards are given for achievements in supercomputing. The three categories are performance, price/performance and compiler parallelization. The last six Gordon Bell Awards are summarized in Fig. 8. They provide a wealth of information about the current trends in parallel computing.

In Fig. 8, the winners of the price/performance category are marked with black rectangles and the winners of the performance category by black circles. After initial successes of SIMD machines (please note three CM2 machine winners in 1989-90) came the reign of Intel machines (Intel hypercube iPSC in 1990, DELTA in 1992 and Paragon in 1994). The price/performance category is clearly dominated by workstations. A quick glance through applications indicates that scientific computing is still the dominant and favoured domain. The trend is very clear in both categories and it indicates rapid exponential growth in the capabilities of the machines.

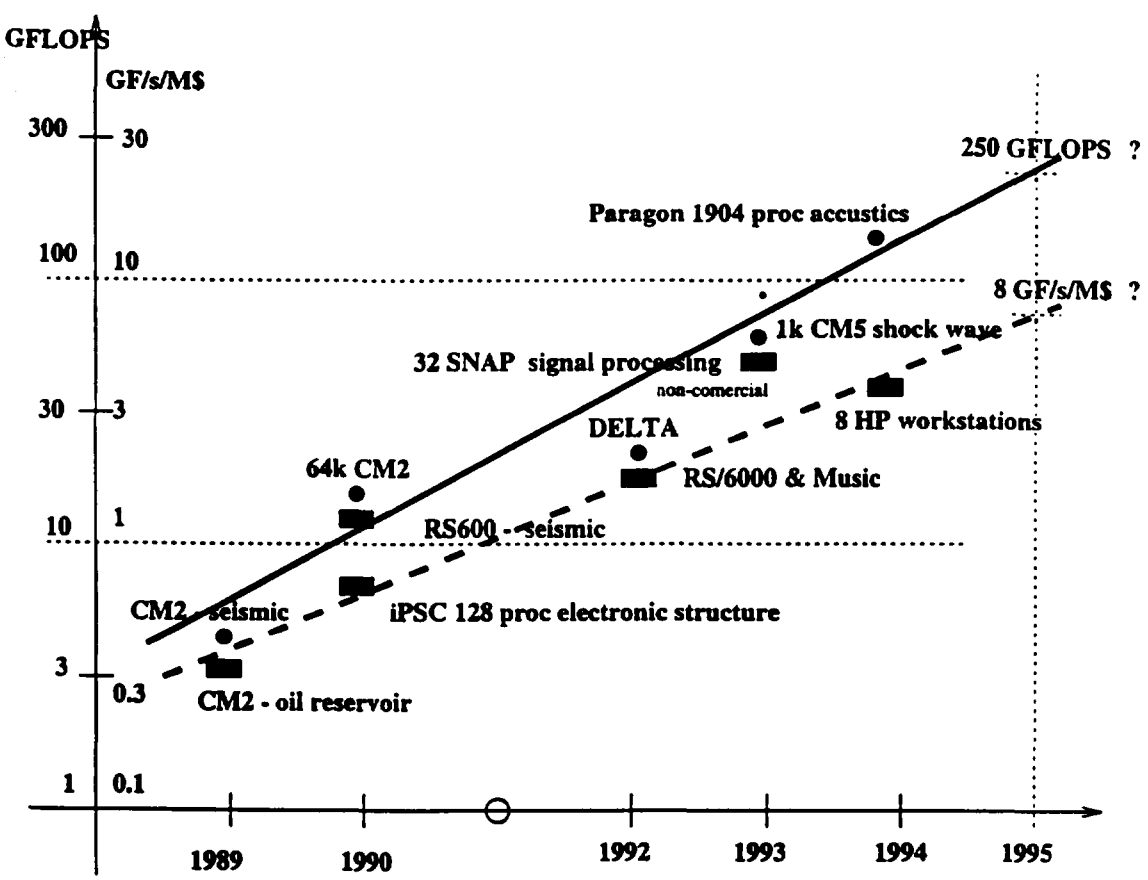


Figure 8. Trends in Gordon Bell Awards Winning Entries.

Over the years, the actual speed record was rapidly growing and the holders of the record were changing quickly. Until recent years, the record was usually held by vectorizing supercomputers with a single or a few processors. Only recently have massively parallel machines started to provide better performance. As of this writing, Intel's Paragon XP/S MP Supercomputer is the record holder, with a sustained speed of 281 GFlops, achieved on an industry standard MP Linpack benchmark in December 1994. The previous record of 170Gflops was made by Fujitsu in August 1994. The Intel machine even achieved 328 Gflops while executing a double-precision complex LU factorization code. The Paragon system used for the record-breaking runs was created by joining two machines at Sandia National Laboratory in the USA. It included 2,256 computer nodes, each with three Intel i860 XP processors, a total of 6,768 microprocessors.

It should be noted that since 1993, awards similar to the USA's Gordon Bell Prizes were introduced in Europe. The so-called SuParCup's are awarded yearly at the Mannheim Supercomputer Conferences.

Distribution of Parallel Programming Resources

Distribution of parallel machines and supercomputers is still heavily concentrated in the most developed countries. The data quoted below are based on the TOP500 list of the most powerful computers in the world, compiled by Dongarra, Meurer and Strohmeir for November 1994. The list publishes Rmax, the maximum performance of a machine on one particular benchmark, so it is not indicative of the speed which can be achieved on an entire application. The total performance installed worldwide in these 500 machines is over 2,600 Gflops, or 2.6 Tflops. The geographical distribution of the computers is given in Table 3

Table 3: The distribution of powerful computers throughout the world

Category	USA & Canada	Japan	Europe	Other Countries
Number of computers	248	82	143	27
Installed power	54%	27%	17%	2%
Leading countries	USA	Japan	Germany, UK	Korea, Australia

A similar list of nearly 200 sites with the most powerful computers is maintained by Gunter. The summary of this list for May 1995 is given in Table 4.

The sites outside of the USA, Europe and Japan were located in Canada, the Republic of Korea and Taiwan, Province of China, (for sites ranked 26-100) and two sites in Australia, two in Hong Kong, as well as single sites in Canada, China, Mexico and Saudi Arabia in tier 100-200. It is interesting to observe that the most powerful sites are mainly governmental laboratories, medium sites are mainly commercial and the smallest sites are mainly academic.

Table 4: The Distribution of Powerful Computing Sites Worldwide

Category	USA	Japan	Europe	Others	Government	Academia	Industry
Sites 1-25	16	7	2	0	14	5	6
Sites 26-100	31	15	26	3	30	15	30
Sites 100-190	40	10	33	8	14	43	34

The growing importance of parallel computing to many countries in the world was demonstrated in the special session of Supercomputing'93, entitled Supercomputing Around the World. Among others, researchers from Indonesia, Malaysia and Singapore were talking about their countries' support for using parallel computers in aerospace, oil and environmental applications.

Software Models

The increasing importance of parallel processing prompted growth in the body of standardization in parallel programming languages and tools. Yet there is no evidence of convergence of the supported programming paradigms to a single model. Currently there are two most popular models for parallel program design: data parallelism and message passing.

Data parallelism is popular because of its simplicity. In this model, a single program (and therefore a single thread of execution) is replicated on many processors and each copy operates on a separate part of data. Depending on the tightness with which the execution of programs is synchronized, there are two modes of using data parallelism. When each instruction of the program is synchronically executed on all processors, then the Single Instruction Multiple Data (SIMD) mode is used. Such tight synchronization requires hardware support.

SIMD machines were quite popular at the turn of the last decade (see Gordon Bell Awards in the previous section). From the software engineering point of view, SIMD machines are easy to program, because there is a single flow of control on all processors. The main focus of parallelization is to find large data structures that can be distributed to all processors to keep them all occupied. Another concern is to minimize the data movements necessary to provide data to processors that are to execute them. Due to the small granule of parallelism (single instruction), SIMD machines consist of a very large number of simple processors (tens or hundred thousands of processors in a single machine is not unusual). Each of these processors must either execute the same statement as all the others or idle, so SIMD machines achieve poor efficiency on programs that do not contain sufficiently large data structures. They also do not perform well on programs that require irregular data references (list structures, dynamic memory, etc.). The consensus is that SIMD architecture has a very specialized niche of applications (e.g., visual information and scene processing), but it is not the best choice for general parallel processing.

Data parallelism can also be used in a loosely synchronized mode, when the program execution consists of two stages:

1. Computational stage, when copies of the same program are executed locally in parallel on each processor. The execution can differ in the conditional branches taken, number of loop iteration executed, etc.,

2. Data exchange stage, when all processors concurrently engage in exchanging non-local data.

It should be noted that the data exchange stage is very simple in the case of shared memory machines (when it can be enforced by use of locks or barriers). The frequency of synchronization in the SPMD model can be adjusted to correspond to the latency of the interconnection network. The SPMD model is quite adequate for scientific computing, which often requires applying basically the same algorithm at many points of the computational domain. SPMD parallel programs are conceptually simple, because of a single program executing on all processors, but more complex than SIMD programs.

For more complex applications, running a single program across the parallel machine may be unnecessarily restrictive. In particular, dynamically changing programs with unpredictable execution times result in poorly balanced parallel computations when implemented in SPMD mode. This is because in SPMD mode, processors synchronize at the data exchange stage, and none of the processors can proceed to the next computational stage until all others reach the data exchange stage.

The SPMD model was abstracted into a Bulk-Synchronous Parallelism model proposed by Leslie Valiant of Harvard University [14]. The model attempts to provide the abstraction for parallel algorithm description that lends itself to performance analysis. The model also became the basis for a library that facilitates the creation of portable parallel software.

The BSP model consists of three components:

1. Processors perform processing or memory functions.

2. A router provides point to point communication between pairs of components.

3. A synchronization mechanism synchronizes all or a subset of the components at regular intervals of L time units (L is called also the synchronization periodicity).

In the BSP model, computation consists of a sequence of supersteps. In each superstep, a component performs some local computation and transmits messages to other components. After a period of L time units, a global check is performed to determine if all components completed the superstep. If not, the superstep is extended by another L time unit, after which the check is made again. In the BSP model, the data transmitted are not guaranteed to be available at the destination until after the end of the superstep at which they were sent.

Using this model, the cost of an algorithm can be expressed in terms of L and g, two parameters that are defined by the network latency and bandwidth, respectively. Using the BSP cost of an algorithm, it is possible to predict the performance of the algorithm on new hardware, given the values of the parameters L and g for this hardware. The BSP model facilitates an algorithm optimization through data distribution selection based on the characteristics of the problem rather than the architectural features of the target machine.

A BSP computer is characterized by the following set of parameters: number of processors p, processor speed s, synchronization periodicity L, and a parameter to indicate the global computation to communication balance g. The synchronization periodicity L is the smallest number of time steps between successive synchronization operations. Parameter g is the ratio of the total number of local operations performed by all processors in one time unit to the total number of words delivered by the communication network in one time unit. Processor speed s is measured in flops (floating point operations per second). Synchronization parameter L is measured in flops. Parameter g is measured in flops per word.

BSP parameters allow for algorithm performance analysis. For example, consider a superstep that needs to communicate h words of data. Since it takes $g*h$ time units for the communication network to deliver the data to its destination, and L units to synchronize all the processors performing the superstep, at least $L+g*h$ units of computation are needed to keep the processor busy; a level of computation less than this threshold results in idling of some processors, and is therefore a source of inefficiency.

In terms of the BSP parameters, distributed memory parallel machines are often characterized by large values of s (relatively fast processors) and low values of L and g (a communication network with low latency and large bandwidth). A general purpose network of workstations, on the other hand, is characterized by values of s that are somewhat lower than for the parallel machines and values of L and g that are much larger than the corresponding values for the parallel machines (high latency and low bandwidth due to the loosely coupled nature of these networks). Thanks to this distinction, optimal BSP algorithms for networks of workstations use different data distribution than those designed for a parallel computer.

BSP algorithms can be directly implemented in a high-level traditional language (e.g., C or Fortran) with the addition of the necessary calls to BSP primitives. The Oxford BSP Library [9], developed by Richard Miller, can be used for this purpose. The library is based on a slightly simplified version of the model presented in [14]. These simplifications require that the processors are allocated statically before the program is run and the programs are written in a SPMD mode. The most significant feature of the library is the support for remote assignment as a means for non-local data access. The library consists of just six functions and is simple to use. Despite its simplicity, we have found it to be quite useful and robust. The library for C-BSP programming includes the functions for

1. Starting and ending a BSP session,

2. Starting and ending a superstep,

3. Fetching and storing a values from a remote processor.

Computationally intensive applications with frequent communication and synchronization require careful design for efficient execution on networks of workstations. Such design is supported by the Bulk-Synchronous Processing (BSP) model. In [11], the authors demonstrate the implementation of a plasma simulation on a network of workstations and the use of BSP analysis techniques for tuning the program for this kind of a machine. They also compare the performance of the BSP implementation with a version based on MPI and conclude that the BSP model, serving as the basis for an efficient implementation, compares favourably with MPI.

The memory distributed machines use message passing to exchange data between different processors. The SPMD model may shield the user from specifying the detailed data movements, thanks to data distribution directives from which a compiler generates the message passing statements. However, the user who decides to write the message passing statements himself has full control over the program execution. In particular, the user may define when and how many processors synchronize in their execution. This gives the user a lot of flexibility at the cost of requiring the user to make a very intricate and detailed description of the program. The programs tend to be longer and more complex than their SPMD counterparts, and therefore more error prone. Once debugged and tuned up, they are also more efficient. The flexibility of the message passing model makes it applicable for a wide variety of problems. As discussed below, the newly developed standard library of functions for message passing, MPI, has the potential of becoming a universal tool for parallel software development.

Trends in Languages

There is a plethora of research parallel programming languages with different flavours to choose from, starting from functional, dataflow to object oriented, logical, etc. However, the majority of parallel programs are still written in Fortran. Since the 1950's, this language has been a favourite choice of writers of scientific programs and particularly for generations of graduate students in the applied sciences. Over the years, Fortran underwent a remarkable transformation, from one of the first languages ever, to the first language with a well defined standard (Fortran66), to the structured programming of Fortran77, to data parallel and object-oriented Fortran90, and finally to the newest standard of High Performance Fortran (HPF). Each generation brought with it new features and set a new standard for the manufacturers of hardware and compilers.

Compared to Fortran77, Fortran90, which was introduced at the beginning of this decade, brought to the world of Fortran users several modern language design features, such as:

1. Derived types, kinds, pointers and dynamic memory allocation that enable users to define their own data types and dynamically allocate data structures.
2. Modules, characterized by public and private data types. Modules can be imported from other programs by the USE clause and renaming.
3. Array operations and new control structures allowing for a very concise and elegant definition of data parallel programs.
4. Recursive procedures.
5. Interface blocks for abstract definition of the input/output, as well as terminal-oriented source forms.

The first two features enable the users to write object-oriented programs. In brief, object-oriented programming involves developing the user's own abstraction of the application domain. This abstraction is defined by the user in the form of data abstraction, object types and type inheritance. An object is defined by its (hidden) state and a set of operations that are applicable to it. Abstract data type is just a set of objects, whereas a class is an abstraction of objects. Each object has private data and attributes that define its implementation, and public data and attributes visible to users of the object. Such a distinction between the object's data and attributes is often referred to as data encapsulation. Finally, polymorphism and function overloading are other characteristics of an object oriented language. Basically, they allow an operator or a function to carry different processing for different types of their arguments. The simplest example of such overloading is its use to define an optimized function of raising to a power. It could be done by using a power series approximation for non-integer exponents and also by an iterative multiplication for integer exponents. Careful analyses of Fortran90 features indicate that all the above features can be expressed in Fortran90 [10].

Another important feature of Fortran90 is the ability to operate on the whole arrays. Array expressions allow the user to define arrays of various shapes and apply operators to such arrays in a piecewise manner. Array shapes can use a set of conditions to decide to which particular elements of the argument an operation should be applied (WHERE clause). The array expressions allow for a very succinct definition of data parallel operations.

A new generation of parallel Fortran, HPF, was introduced in 1993 by an HPF Forum, a group of parallel hardware manufacturers and academic, industrial and governmental users of high performance machines. During 1994 there were six announced commercial HPF products and 11 announced commercial HPF efforts, with many of these compilers becoming available by mid-1995. HPF introduced new data partitioning directives, such as ALIGN/REALIGN data structures relative to each other, DISTRIBUTE/REDISTRIBUTE data structures (or their templates) to processors according to one of the predefined patterns (BLOCK, CYCLIC, or BLOCK-CYCLIC). Directives for definition of processor arrangements (PROCESSORS), or loop parallelization (INDEPENDENT/FORALL) are available.

These directives enable the user to define data movements indirectly without the need for a detailed description of the message passing statements that must be executed to achieve a directive-defined effect.

Critics of HPF think that the HPF standard is not general enough. In particular, HPF does not allow for dynamically defined alignments and distribution that are permitted in Fortran HPF+ [3]. However, standardization of the language features is extremely important for users, compilers and tool writers, because it protects their software investments against changes in the architecture. In that respect, the introduction of Fortran90 and then HPF was an important step forward towards more stable parallel software.

HPF can be seen as the flagship of the data parallelism camp. On the other hand, the supporters of message passing based parallel programming achieved standardization of their approach in the Message Passing Interface (MPI). MPI is a large library of the message passing utilities that includes 125 functions. The basic MPI subset, sufficient for writing simple applications, consists of just the following six functions:

1. MPI_INIT – to initialize MPI on in a process,
2. MPI_COMM_SIZE – to find the number of processes participating in the MPI session,
3. MPI_COMM_RANK – to find a unique rank of the calling process among the MPI session participants,
4. MPI_Send— to send a message to the other processes,
5. MPI_Receive – to receive a message,
6. MPI_Finalize – to terminate the MPI session.

The innovations of MPI are centred around an abstract view of the communication. This abstract view supports the portability of programs using MPI to different machines. Messages in MPI are described as triples, consisting of an address, data count and data type. Data types in such triples can be user defined. MPI allows the processes to group themselves and arrange themselves into a hierarchy where each process has its own rank. A process can have different ranks inside different groups and it can participate in different communication sessions concurrently. MPI also provides a default initial group whose members are all processes that executed the MPI_INIT function. Families of messages can be defined in terms of communication context and group.

MPI also provides more complex features, such as collective communication that includes data movements and global reduction operations. MPI allows the user to define virtual topologies and use different communication modes. It also provides functions for debugging and profiling, and support for heterogeneous networks. The MPI standard does not define, purposefully, how the MPI startup is implemented, the amount of system buffering,

or the limitations on recognized errors, to avoid unnecessary restrictions on implementations.

Judging from the widespread popularity of the Parallel Virtual Machine (PVM), MPI can become an important step towards providing an efficient and unifying tool for expressing message passing in parallel and distributed applications. Although introduced recently (in 1993), MPI has been quickly embraced by manufacturers and is supported on many parallel machines (among them the Cray T3D, Intel Paragon and IBM SP2).

Conclusions

Parallel processing is at a critical point of its evolution. After a long period of intense support by government and academia, it slowly moved to derive the bulk of its support from the commercial world. Such a move brings with it a change of emphasis from record breaking performance to price performance and sustained speed of program execution. The winning architectures are not only fast, but also economically sound. As a result, there is a clear trend towards widening the base of parallel processing both in hardware and software. On the hardware side, that means using off-the-shelf commercially available components (processors, interconnection switches), which benefit from the rapid pace of technological advancement, fueled by the large customer base. The other effect is the convergence of different architectures, thanks to spreading the successful solutions among all of them. Workstations interconnected by a fast network approach the performance of parallel machines. Shared memory machines with multilevel caches and sophisticated prefetching strategies execute programs with an efficiency similar to distributed memory machines.

On the software side, widening the base of users currently relies on standardization of parallel programming tools. By protecting the programmer's investment in software, standardization promotes development of libraries, tools and application kits that in turn will attract more end-users to parallel processing. It appears that parallel programming is ending a long period of craft design and is entering a stage of industrial development of parallel software. This is an industry in the making that will provide new opportunities for software developers and investors.

Acknowledgment

Some of the research discussed in this section has been supported from the following US agencies: Office of Naval Research under N00014-93-1-0076, National Aeronautical and Space Agency under NGT-70334, and National Science Foundation under grants ASC-9318184 and BIR-9320264. The content of this section does not necessarily reflect the position or policy of the US Government – no official endorsement should be inferred or implied.

References

[1] Gordon Bell, *"Why there won't be apps: The problem with MPPs,"* **IEEE Parallel & Distributed Technology: Systems & Applications**, Vol. 2, No. 3, pp. 5-6, 1994.

[2] Philip Carnelley and William Cappelli, *Beyond the Data Warehouse, New Markets for Parallel Computing,* **Ovum Reports**, 1995.

[3] Barbara Chapman, Hans Zima and Piyush Mehrotra, *"Extending HPF for Advanced Data-Parallel Applications,"* **IEEE Parallel & Distributed Technology: Systems & Applications**, Vol. 2, No. 3, pp. 59-70 , 1994.

[4] James Cownie, *"Why MPPs?"*, **IEEE Parallel & Distributed Technology: Systems & Applications**, Vol. 2, No. 3, pp. 7-8, 1994.

[5] Evolving High Performance Computing and Communications Initiative to Support the Nations's Information Infrastructure, National Research Council, National Academy Press, Washington, D.C. 1995.

[6] John L. Gustafson, *"Reevaluating Amdahl's Law"*, **Communications of the ACM**, Vol. 31, No. 5, pp. 532-3, 1988.

[7] John Hennessy and David A. Patterson, Computer Architectures: A Quantitative Approach, Morgan Kaufman Publishers, Inc., 1990.

[8] Ken Kennedy, *"Is Parallel Computing Dead ?"*, **Parallel Computing Research**, Vol. 2, No. 4, pp. 2-19, 1994.

[9] Richard Miller and J.L. Reed, The Oxford BSP Library: Users' Guide, Version 1.0, **Oxford Parallel Technical Report**, Oxford University Computing Laboratory, 1994.

[10] Charles Norton, Boleslaw K. Szymanski and Viktor Decyk, *"Object Oriented Parallel Computation for Plasma PIC Simulation"*, **Communications of the ACM**, Vol. 38, No. 9, to appear, 1995.

[11 M. Nibhanupudi, C. Norton, and B.K. Szymanski, *"Plasma Simulation on Networksof Workstations using the Bulk-Synchronous Parallel Model"*, **Proc. International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '95)**. Athens, GA., November 1995, CSREA, 1995, pp. 13-22.

[12] Monica Snell, *"Supercomputing: Same name, different game"*, **IEEE Computer**, Vol. 27, No. 11, pp. 6-7, 1994.

[13] Steve Wallach, *"Teraflops into laptops"*, **IEEE Parallel & Distributed Technology: Systems & Applications**, Vol. 2, No. 3, pp. 8-9, 1994.

[14] Leslie Valiant, *"A bridging model for parallel computation"*, **Communication of the ACM**, Vol. 33, No. 8, pp. 103-11, 1990.

[15] David Wood and Mark Hill, *"Cost-Effective Parallel Computing"*, **IEEE Computer**, Vol. 28, No. 2, pp. 69-72, 1995.

Glossary

ATM: Asynchronous Transfer Mode, a new standard of transmitting data over a network that unifies the needs of computer processing and telecommunications (voice and video transmission).

BSP: Bulk Synchronous Parallelism model developed to unify algorithm description for parallel machines.

Cache: Fast but expensive memory used to speed up access to data in main memory of the computer.

Cluster of Workstations: or COW, a parallel machine created by joining independent workstations by a network (usually Local Area Network, LAN).

DRAM: Dynamic Random Access Memory, is currently the technology producing the densest computer memory chips (16-64MB or millions bytes in a single chip).

File server: A special computer in a network of workstations responsible for providing the file storage and services for the entire network.

Massively parallel machine: A computer with many processors, not necessarily the fastest computer on the market (see supercomputer).

Multimedia: Use of numerical data, voice and pictures/movies, in data processing.

Multiprocessor: A computer with many processors, a synonym of parallel computer. Supercomputers are now often multiprocessors.

SIMD machine: Single Instruction Multiple Data multiprocessor. It consists of a large number of simple processors, each executing the same instruction.

SPMD: Single Program Multiple Data mode of parallel processing. Each of many processors executes the same program on different data. Unlike the SIMD computer, conditional statements may cause that at any given instant each of the processors in SPMD mode may execute different instructions.

Supercomputer: Ultra-fast computer for numerical computation, usually based on vector units (specialized processors for matrix and vector operations) and some, not necessarily massively, parallelism.

Uniprocessor: a computer with a single processor, a synonym of sequential machine. The first supercomputers were uniprocessors with vector units.

III. High Performance Computing in India and the Far East

by

L.M. Patnaik

Summary

With a vision for developing its own advanced computing technology based on strong intellectual resources, India launched a major initiative in parallel computing in 1988. The launch of the Centre for Development of Advanced Computing (C-DAC) and concurrently other efforts at the National Aerospace Laboratory (NAL), Bangalore; Advanced Numerical Research & Analysis Group (ANURAG), Hyderabad; Bhabha Atomic Research Centre (BARC), Bombay; Centre for Development of Telematics (C-DOT), Bangalore, marked the beginning of high performance computing in India. Today, India has designed its own high performance computers in the form of PARAM by C-DAC, FLOSOLVER by NAL, PACE by ANURAG, ANUPAM by BARC and CHIPPS by C-DOT. These machines, being contemporary in architecture, brought down the cost of equivalent international machines in the Indian market place, created a high performance computing culture in India, and contributed to several mission critical advanced research programmes. C-DAC is now on the way to developing its own massively parallel teraflops architecture by 1998, placing India in league with several advanced countries, as far as supercomputing technology and applications are concerned. This chapter reviews these developments in India, apart from highlighting the future directions in this area. A brief discussion on the high performance computing activities in some other countries of the Far East is also included.

Introduction

Although the performance of single processors has been steadily increasing over the years, the only way to build the next generation teraflop architecture supercomputers seems to be through the root of parallel processing.

Even with today's workstation class, high-performance processors crossing 100 mega-flops, thousands of processors are required to build the teraflop architecture machine. One of the grand challenges of the computing technology of the 1990s is to build high-performance computers in the teraflops range. The high-performance computers are required to

solve the so-called grand challenge problems of science and engineering. Several such grand challenge problems have now been posed and the number is continualy growing.

From the applications point of view, there are several motivations to building teraflop machines to solve grand challenge problems in science and engineering. These grand challenge problems include designing aircraft, simulating semiconductor materials, analyzing fuel combustion, rational design of drugs, understanding catalysis, designing protein structures, researching the human anatomy, imaging, predicting the weather, studying air pollution, ocean modelling, evaluating ozone depletion and oil exploration. The benefits that result from the application of these high speed machines will touch every facet of life, such as the air we breathe, the weather, the ozone and global-warming problems that are threatening humankind, design of new generation aircraft, drug research, improved medical therapies, more efficient engines and cars, new chemicals, stronger structural materials, and faster VLSI chips. The goal is to model reality more and more accurately on the supercomputers and to simulate the real phenomena in unprecedented details of chemistry, physics, biology and other sciences. Supercomputing has now become the third mode of scientific investigation in the 1990s, complementing theory and practice. It is now widely believed that the supercomputing technology, at its cutting edge, will hold the key to the future competitivity of nations in advanced science and technology, business and security.

In 1987, India decided to launch a national initiative in supercomputing, in the form of a time-bound mission to design, develop and deliver a supercomputer in the gigaflops range. The major motivation came from the delays in obtaining a CRAY XMP for weather forecasting on the one hand, and a firm belief that India can develop its own high-performance computing technology leveraging upon its high-class intellectual resources on the other. Right from the beginning, it was clear that the destiny of supercomputing in India would be carved through the parallel processing route. The Centre for Development of Advanced Computing (C-DAC) was set up for this purpose in August 1988 with the First Mission three-year budget of Rs. 375 million (approximately US$ 12 million). The launch of C-DAC marks the beginning of high performance computers in India.

The only high performance computers in India at that time were the IBM 3090, DEC 10/20, CDC Cyber 730/830, ICL 2100, NEC S1000 series mainframe computers, NORSK DATA ND1000/550 superminis and HP 1000, DEC PDP 11 and MicroVAX minicomputers. The peak computing power of these mainframe computers, superminis and minis was less than 10 megaflops. Supercomputers were viewed as a strategic resource for India's advanced education and research programme.

C-DAC's First Mission was directed to deliver 1000 Mflops parallel supercomputers by 1991. Simultaneously, several other complementary projects were initiated to develop high-performance parallel computers at the National Aerospace Laboratory of the Council of Scientific and Industrial Research (CSIR), the Centre for Development of Telematics (C-DOT), the Advanced Numerical Research & Analysis Group (ANURAG) of Defence Research and the Development Organisation (DRDO) and Bhabha Atomic Research Centre

(BARC). India's first generation parallel computers were delivered, as of 1991. The C-DAC has already achieved significant commercialization of its advanced computing technology. India is now in the process of announcing its second generation parallel computer and the C-DAC has already begun its Second Mission to develop teraflop architecture massively parallel supercomputers. This article presents the current state-of-the-art of Indian high performance computers which arose from the efforts of the C-DAC, BARC, NAL, ANURAG and C-DOT.

C-DAC's PARAM 9000

Background and Evolution

C-DAC formally launched its first mission in August 1988 to deliver the gigaflops range parallel machines. C-DAC had started almost from scratch with very little knowledge in this fast advancing field, but under the dynamic leadership of its executive director, Dr. V.P. Bhatkar, it produced the first 64 node prototype in barely two years. The machine was taken to Zurich for CONPAR 90, a major parallel processing conference in Europe, and the international scientific community saw that India had developed a parallel computer comparable to the machines developed in Europe. Indeed, the target machine of 256-node complete with parallel disk array storage, comprehensive parallel programming environment and multi-user and multi-host capability was delivered in August 1991 without time or cost overruns.

C-DAC's parallel supercomputers have been named PARAM, meaning "Supreme" in Sanskrit. It also made a nice acronym for a PARAllel Machine. The programming environment is called PARAS (the mythical stone which can turn iron into gold by a mere touch), which gave a golden touch to the underlying machine and made the job of programmer or user reatively easy. The first PARAM series scalable supercomputers were based on INMOS Transputers 800/805 as computing nodes, and the first PARAM models were called PARAM 8000 series systems.

During the first mission, C-DAC launched a fairly large applications development programme spanning more than 20 academic institutions, research laboratories and industries. Through a series of workshops, over 200 scientists and engineers were trained to use PARAM. A lot of hand-holding was done to parallelize the sequential codes on PARAM. Initial results were stunning, the performance of PARAM 8000's single node sometimes exceeded the then performance of the available popular workstations such as microVAX, and even mainframes. The scalability of PARAM 8000 was demonstrated in a variety of applications, for example Finite Element Methods, Computational Fluid Dynamics, Computational Physics, Computational Chemistry and Monte Carlo Simulation.

Although the theoretical peak-performance of the 256 node PARAM machine was one gigaflop (single node T805 claiming 4.25 Mflops), its sustained performance in actual application turned out to be between 100 to 200 Mflops. The scalability in many applications however, was excellent.

At the beginning of 1992, the basic compute node of PARAM 8000 was found to be underpowered. INTEL had announced their i860 RISC processor, which claimed a theoretical peak-performance of 60 Mflops. A decision was taken at this time to integrate i860 into the PARAM architecture. C-DAC's objective was to preserve the same application programming environment and provide straightforward hardware upgradability by just replacing the compute node boards of PARAM 8000. This resulted in the next architecture with i860 as a main processor with four transputers acting as communication processors, each with four built-in links.

PARAS programming environment was extended to PARAM 8600 to give an identical user view as PARAM 8000. During 1992 and 1993 C-DAC succeeded in building a scalable parallel machine, which was called PARAM 8600.

The computing power of four compute clusters of PARAM 8000 could now be realized in a single compute cluster of PARAM 8600. The sustained performance of the 16 node PARAM 8600 was in the range of 100-200 Mflops, depending on the application. Thanks to PARAS 8600, PARAM 8000 applications could be easily ported on PARAM 8600 machines.

The effort required to deliver the PARAM 8000/8600 machines exceeded 300 man years. Everything was developed from the root level, only the chips were imported. The software effort extended to over a million lines of source code! In this process, C-DAC built strong foundations to undertake the next challenges.

In August 1988, C-DAC began from scratch with a small core team; at the end of 1992, a strong institution of 200 outstanding scientists and engineers was created with its headquarters at Pune and centres at Bangalore and New Delhi, India. C-DAC had also installed a world class Electronic Design Automation (EDA) and software development environments. The institution became well-known, not only in India, but also throughout the world. Several interational alliances were formed with leading international universities from Russia, Europe and North America.

Second Mission

It is against this backdrop of the accomplishment of the First Mission that C-DAC started its Second Mission. While in the First Mission C-DAC delivered the gigaflops range parallel supercomputers, the goal of the Second Mission was to deliver the teraflops range massively parallel grand challenge supercomputer. This meant that C-DAC was aiming for a supercomputer with a 1,000 times more performance than that which had been delivered in the

First Mission. The motivations for this goal were many. C-DAC wanted to consolidate the gains of the First Mission and run the race for the teraflops range of supercomputers. India became the only country running this race outside the USA, Europe and Japan.

Within 18 months of its Second Mission, C-DAC announced its second generation machine called PARAM 9000. The machine was exhibited at Supercomputing '94, Washington. Progressively through the succeeding years, PARAM 9000 will be scaled up to the teraflops level.

System Architecture

C-DAC has advented the OpenFrame Architecture for the PARAM 9000 series systems, which heralds the era of flexible supercomputing. The OpenFrame Architecture brings a new and innovative concept in scalable parallel supercomputing. The distinctive features of the architecture are processor independence and unification of cluster and massively parallel computing. The new architecture allows the integration of different processors as compute nodes and unifies both cluster and massively parallel computing within a single framework, thus providing a truly open framework. The inherent property of OpenFrame Architecture permits continous technology upgrading as new processors and advances in interconnect technology become available.

The OpenFrame systems support the popular and powerful RISC processor based compute nodes, industry standard networking, I/O devices, standards based parallel programming environment. The OpenFrame Architecture provides for a variety of scalable I/O and networking interfaces, with Ethernet and fast and wide SCSI as standard. Available options are multiple HiPPI, multiple FDDI, multiple SCSI, Ethernet and ATM interfaces.

The nucleus of the OpenFrame Architecture is a modularly scalable multistage interconnect network accommodating more than a thousand heterogeneous processing nodes. The internode communication is via a low latency, high bandwidth point-to-point link. The multistage interconnect network of PARAM 9000 uses a packet switching wormhole router as the basic switching element. Each switch is capable of establishing 32 simultaneous non-blocking connections to provide a sustainable bandwidth of 320 MBytes/sec. The communication links of PARAM 9000/SS conform to the IEEE P1355 standards for point-to-point links.

The interconnect network is non-blocking and provides full connectivity to all the nodes, thus freeing the programmers from having to concern themselves with topologies. All the nodes are equidistant, resulting in predictable and repeatable performance. The dynamic adaptive routing ability of the interconnect network avoids hot spot build-up, which is typical in irregular problems. If the targeted output link is busy, other links in the group are dynamically and automatically selected for message routing, thereby removing the congestion. Future additions to the interconnect network include hardware support for HPF and MPI. Planned features are for the global reduction, broadcast/multi communications, barrier

synchronization and data distribution. The communication fabric allows the partitioning of the system into production and development environments in any arbitrary manner and can be changed by the system administrator to suit the dynamic needs. Even within these two partitions, multi-user access is provided and these partitions are non-blocking.

The PARAM 9000 carries the philosophy of flexibility into the node architecture. As new technologies in processors, memory and communication links advance and become available, these can be upgraded in the field. The first offering of PARAM 9000 architecture is the PARAM 9000SS system based on SuperSparc series processors. The complete node is realized using the SuperSparc II processor with 1 MB of external cache, 16 to 128 MB of memory, one to four communication links and related I/O devices. The current operating speed of the processor is 75 MHz and as and when new MBus modules with higher frequencies become available, they can be field-upgraded. MBus speed is 50 MHz. Each node can be configured with 16, 32, 64, 80 or 128 MB high speed, error correcting memory and can be field-upgraded.

Communication links can be scaled from one to four links, providing a bandwidth of 10, 20 or 40 MBytes/sec. The processor is used only for the initiation of the message and is interrupted whenever a programmable number of messages has been received. The DMA engine on the communiction interface performs the message packetization, routing of packets on all the available links and reassembly of packets. The communication protocol defines three message types based on the message length; very short message, short message and long message. While the very short message is typically used by the kernel and can be assigned priority over others, the other two types of messages are packetized and time-multiplexed.

One out of every four nodes can be configured as an I/O or server node, satisfying a variety of I/O needs. While the service nodes run the Solaris operating system, the compute nodes run the PARAS microkernel.

Users can now integrate the SPARC workstations into the PARAM 9000/SS by just adding the SBus based network interface card. Each network interface card supports one, two or four communication links. C-DAC also provides the necessary software drivers.

The OpenFrame Architecture supports a variety of scalable networking and mass storage options conforming to industry standards. These interfaces are connected to the I/O nodes running the Solaris operating system. PARAM 9000/SS supports multiple fast and wide SCSI channels for connection to external RAID boxes. Any off-the-shelf SCSI-in-SCSI-out RAID boxes can also be used to realize a range of storage needs, including C-DAC's own scalable I/O system. Through the Solaris operating system, the user can perform all the regular file operations, including NFS, remote file system, etc. The system's I/O nodes support standards conforming interfaces to Ethernet, FDDI and HiPPI. Via the Solaris operating system, the standard software utilities and protocols such as FTP, TCP/IP and sockets, telnet, NFS, etc. are supported.

Programming Environment

A high performance computing system is only as open and flexible as its software. The open and flexible software environment is an integral part of the OpenFrame Architecture advented by C-DAC. The architecture permits the parallel processing system to be viewed as an ensemble of independent workstations, a cluster of workstations or as massively parallel systems connected through a scalable high bandwidth network, or any combination of these.

The users of parallel programming environments are looking for openness, industry standards, a spectrum of sophisticated development tools and flexibility of composition, depending on the requirements for robustness, performance and ease-of-use. The software for OpenFrame Architecture meets this requirement with a comprehensive range of software tools and utilities that can cater to the different needs of the user.

PARAS 9000/SS is C-DAC's parallel program development environment, for C-DAC's Sparc processing nodes based scalable massively parellel system – PARAM 9000/SS. The software environment, PARAS 9000/SS seamlessly blends industry standards with parallel programming extensions to provide both high performance and ease of use. High performance is achieved through an optimized microkernel, parallel high performance file system, standard and enhanced compiler optimizations and parallel libraries.

The PARAS 9000/SS supports the two main massively parallel programming models: data parallelism and multiprocess parallelism. Data parallelism is automatically realized through the support for HPF, while CORE, PVM and MPI message passing interfaces provide the required support for multiprocess parallelism.

Solaris is the preferred program development OS environment of PARAM 9000/SS and runs on all the service nodes of the system. For performance and efficiency during production runs, the PARAS microkernel is replicated on the compute and I/O nodes. The microkernel provides all the necessary services to the application program, while dispensing with the overheads of a standard OS. The result is better performance on the production codes and ease of development through a standard OS. The salient features of the PARAS microkernel are Mach-like process management, enhanced exception handling support, simple virtual memory model and port-based interprocess communication. The abstractions that are supported are tasks and threads, ports and port groups and virtual memory regions. To enhance the portability of applications, popular message passing interfaces PVM and MPI are provided. Applications written using the PVM and MPI libraries can be ported on to PARAM 9000/SS effortlessly.

The operating system configures the system into service, compute and I/O partitions. The user logs onto a service node and uses the PARAS program development tools to develop a parallel application. Upon request from the user, the resource manager allocates a pool of processors and maps the application onto the allocated nodes. The user's parallel application

can be distributed betweeen the compute and service partitions. This flexibility allows the user to configure Solaris servers to provide specialized services to the compute node tasks of the parallel application. The operating system space shares the compute partition across multiple users. The nodes of the service partition run Solaris 2.x while the nodes of the compute and I/O nodes run the PARAS microkernel, along with appropriate servers.

The languages supported by PARAM 9000/SS include ANSI C, C++, Fortran-77, Fortran-90 and the emerging HPF. Either C-DAC's own compilers or third party compilers can be used for program development with standard and enhanced optimizations. The interconnect network of PARAM 9000/SS will provide hardware support for MPI and HPF for speedier execution of the applications. The features that will be supported are global reduction, broadcast/multicast communications, barrier synchronization and data distribution.

PARAS 9000/SS provides a variety of tools for program development and debugging. FORGE 90, the program restructuring tool from Applied Parallel Research, Inc., AIDE and PET from C-DAC are the development tools available under PARAS. C-DAC will also provide a Total View debugger for cluster computing environments soon. PARUL – the parallel libraries from C-DAC contain more than 400 routines for dense linear equation solutions and eigen value determination, sparse linear equation solution, BLAS level 3 – basic linear algebra sub-routines, Poisson's equation solution, 1-D and 2-D FFTs and general purpose sorting.

For flexibility in client/server compute environments, the system's service nodes support standards conforming interfaces for Ethernet and FDDI connections and C-DAC's interconnect network. Support for ATM networks will be added soon. Networking software includes standard utilities and protocols such as ftp, TCP/IP, sockets, telnet and NFS.

In addition to accessing the file system of the service nodes, a separate dedicated and high performance scalable mass storage server based on multiple I/O nodes is also supported to provide high bandwidth and large storage capacity. Parallel applications that run on compute nodes under PARAS can use both UNIX file I/O calls, or parallel I/O primitives, which efficiently and transparently map structured data such as matrices over multiple I/O nodes. Reliability and speed are enhanced by connecting RAID systems to the I/O nodes. The mass storage server follows the IEEE Mass Storage System Reference model.

A distributed visualization environment will be supported on PARAM 9000/SS. KHOROS, one of the most widely used visualization system by computation scientists, will be provided as the basic environment. Other standard visualization packages, such as IRIS Explorer, will also be made available.

The PARAM 9000/SS system can be partitioned mainly into batch, interactive and production partitions.These partitions can be created by the systems administrator, depending on the need. C-DAC supports CODINE, the distributed computing management utility for the management of heterogeneous workstation clusters, which are integrated with vector and parallel compute servers.

C-DAC is committed to maintaining backward compatibility with its earlier generation of the PARAM series of parallel machines. Applications developed using PARAS 8000/8600 can be recompiled and executed on the PARAM 9000/SS machines. Applications developed on workstation clusters and other parallel machines using the PVM and MPI message passing interfaces just need to be recompiled for execution on PARAM 9000/SS.

C-DAC will enhance the PARAS 9000/SS programming environment to provide a single system image of UNIX. The directions for the parallel programming environment are evolving continously and new standards are being defined. The parallel programming environment as envisioned in the OpenFrame Architecture is planned to conform to these emerging standards.

Applications on PARAM

Development and porting of parallel processing applications is a major driving force of the C-DAC Mission. In the First Mission, over 40 application kernels were developed in collaboration with user agencies and demonstrated on PARAM 8000 and 8600 series machines. Emerging application areas, such as parallel database management, complex query decision support systems and video-on-demand, will also be supported. C-DAC's application development programme includes development of parallel libraries, application kernels and benchmarking, parallelization and porting of production quality industry standard packages, *ab initio* development of parallel application packages in select areas, wide-scale catalysis and education and research in parallel processing, and installation and networking of parallel supercomputing facilities.

Several application kernels have been developed on PARAM 8000/8600 series machines in the areas of computation fluid dynamics, finite element analysis, oil reservoir modelling, seismic data processing, image processing, remote sensing, medical imaging, signal processing, radio astronomy, molecular modelling, biotechnology, quantum molecular dynamics, quantum chemical calculations, semiconductor physics, composites and special materials, power systems analysis and energy management, and discrete optimization. In the Second Mission, attention has been focussed on the parallelization and porting production quality industry standard codes in collaboration with various organizations.

BARC's ANUPAM

Background and Evolution

The Bhabha Atomic Research Centre is a premier national centre for nuclear science and allied disciplines founded by Dr. Homi Bhabha, and has since then been at the forefront of

India's Atomic Energy Programme. Over 2,000 scientists and engineers make use of the central computing facility for solving their computational problems. Until 1992, the computers available to BARC scientists were the last generation Norsk Data Computers, which took hours of computational time over moderate problems of BARC scientists and engineers. BARC began some initial work on parallel processing by configuring a transputer network for image processing based on hardware procured from INMOS and C-DAC. Through 1991 and 1992, BARC computer facility members started interacting with C-DAC for a high-performance computing facility. It was estimated that a machine of 200 Mflops ofsustained computing power would be needed to solve the current problems. In view of the vulnerability of this programme, BARC decided to build their own parallel computer.

In 1992 the BARC system integrated their own parallel computer based on the standard Multibus II i860 hardware. Initially, an 8-node machine was announced, which was expanded to 16 node and then to 24 and 32 node systems. BARC's parallel machine was called ANUPAM, meaning "unparalleled" in Sanskrit. BARC transferred the technology of ANUPAM to the Electronics Corporation of India Ltd. (ECIL) located in Hyderabad, a public sector unit manufacturing electronic systems under the umbrella of the Department of Atomic Energy of the Government of India.

System Architecture

BARC's ANUPAM is a MIMD architecture (Fig.1), machine realized through standard off-the-shelf MULTIBUS II i860 cards and crates supplied by WIPRO. Each node is a 64-bit i860 processor with 64 KB cache and a local memory of 16 to 64 MB. The peak computing power of a single node is 100 Mflops, although the sustained power is much less. The first version of the machine was announced with eight nodes in a single cluster (or Multibus II crate). There is no need for a seperate host (Node O acts as the host processor).

ANUPAM is scalable to 64 nodes. The inter-cluster message passing bus is a 32-bit MULTIBUS II backplane bus operating at 40 MB/sec. peak. This bus is shared by eight nodes within a cluster. The communication between clusters is achieved through two 16-bit wide SCSI busses – one in X and the other in Y direction, forming a 2D mesh. The inter-cluster bus realized through standard SCSI controller chips has a peak speed of 20 MB/sec. Standard topologies, such as mesh, ring, hypercube, etc., can be easily mapped on the 2D mesh topology. The interconnection network is scalable to 64 processors with eight MULTIBUS II crates. Each cluster also handles the I/O requirements of the system.

Programming Environment

The ANUPAM parallel programming environment is simple and straightforward. The master processor runs a UNIX SVR4 operating system. The user writes his parallel program in FORTRAN 77 or ANSI C language. A FORTRAN vectorizer is provided. To aid program development and debugging, a profiler and a debugger are provided. The parallel simulator (PSIM), which runs on any UNIX machine, provides a simulated parallel environ-

ment of ANUPAM and helps users in parallelizing and debugging algorithms without the need for a target machine.

The parallel processing paradigm makes use of Hoare's Communicating Sequential Processes CSP model. In this model, the underlying computing system is a collection of concurrently executing sequential processes on multiple processors, communicating with each other via explicit messages. Each processor has its own code, data and stack.

The message passing library consists of send/receive calls to facilitate the writing of parallel codes by calling the library routines either in Fortran or C language. Apart from this, Program Scheduler, Batch Queue Manager and Parallel Library for scientific and image processing routines are a part of the ANUPAM software environment.

The graphics and visualization support is through a standard graphics workstation, such as SGI connected on Ethernet. The image processing software libraries developed at the Computer Division of BARC make use of the client/server model assembly ANUPAM as a server for image analysis and enhancement and an SGI workstation as a client for image display.

Applications

The ANUPAM parallel computer was installed at the BARC Computer Centre, the configuration was scaled from 8 to 32 nodes, along with hardware enhancements in the later models. The parallel computing facility has been used extensively by BARC scientists and engineers as well as by other users from external institutions. More than 50 applications have been parallelized and are being used by users. Studies have also been carried out on algorithms employing data domain decomposition, algorithmic parallelisation, geometric parallelisation, event parallelisation, etc.

For a standard Linpack benchmark with 1000 x 1000 matrix size in double precision calculations, ANUPAM 8 gave 52 Mega Floating Point operations per second (Mflops) employing eight nodes with 80 per cent efficiency. Matrix multiplication with 1000 x 1000 matrix size obtained 106 Mflops performance on the eight node ANUPAM system with 88 per cent efficiency. A molecular dynamic program to calculate electronic structures using linear approximations in BAND theory demonstrated efficiencies of the order of 81 per cent on the 16 node configuration. A program called PROLSQ, to calculate protein structure refinement by the method of least square fit, runs on ANUPAM 8, giving 7.45 times more speed compared to a single node. The computational problems in multidisciplinary aerospace science are also running successfully on ANUPAM 8, ANUPAM 16 and ANUPAM 24 systems, displaying better efficiencies. The VASBI (Viscous Analysis of Symmetric Bifurcated Intake) code used for calculating airflow around an aircraft exhibited 5.4 times more speed than the IBM RS 6000/560 using 24 nodes of ANUPAM (measuring over 160 Mflops).

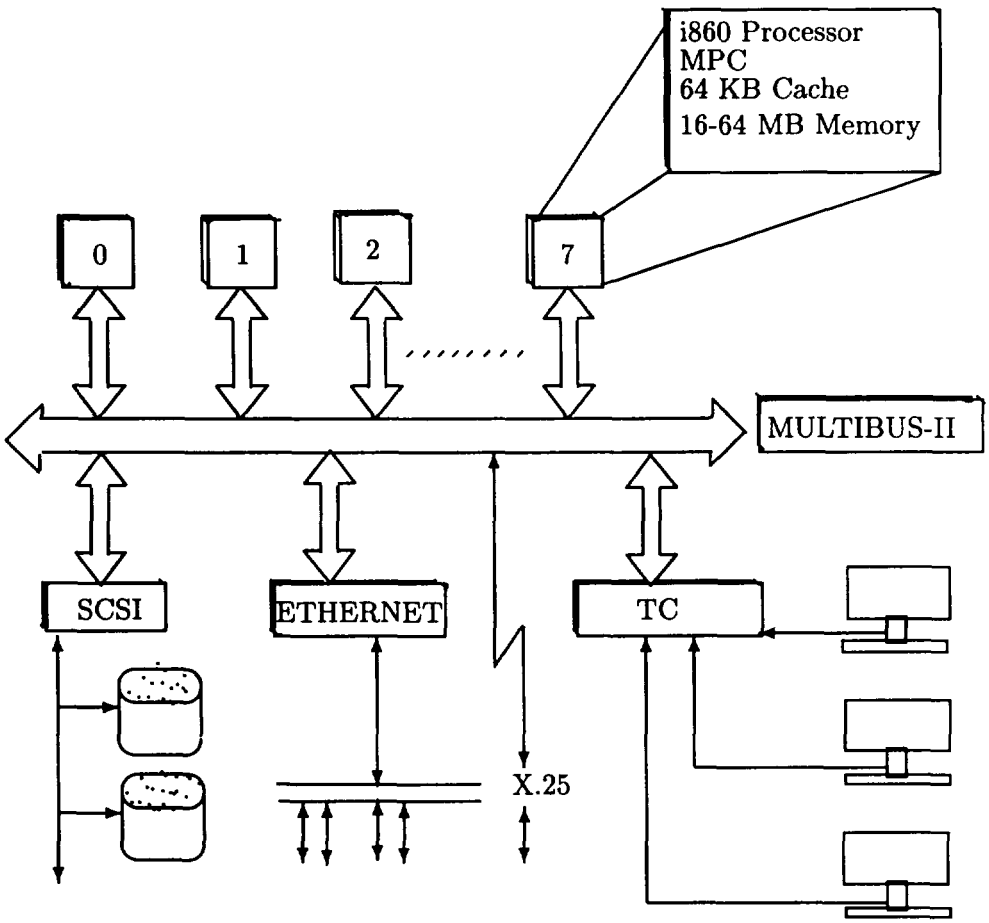


Figure 1: ANUPAM: Single Cluster Architechture

Future

It is understood that the ANUPAM system is evolving with an i860 XP processor with per node memory extended to 128 and 256 MB. The intercluster bus is being expanded to fast and wide SCSIs. An X-Y router chip is planned. The present slow speed Ethernet link to the graphics workstation will be replaced by a high speed graphics adapter sitting directly on a Multibus II. The graphics adapter will make use of an i860/XP graphics processor and will provide TCP/IP support over Multibus II, thus maintaining compatibility with the existing graphics interface.

ANURAG's PACE

Background and Evolution

The Advanced Numerical Research and Analysis group (ANURAG) located in Hyderabad, is a recently created laboratory of the Defence Research Development Organization (DRDO) focussed on R&D in parallel computing and VLSIs and applications of High

Performance Computing in CFD, medical imaging and other areas. ANURAG has developed PACE, a loosely-coupled, message-passing parallel processing system. PACE is an acronym for Processor for Aerodynamic Computations and Evaluation. ANURAG's PACE programme began in August 1988. As its name suggests, PACE was originally designed to cater to the needs of Computational Fluid Dynamics (CFD) requirements of aircraft design. However, since CFD essentially involves the solution of partial differential equations, PACE is also targetting scientific computations.

The initial prototypes of PACE were based on the Motorola MC 68020 processor. The hardware was supplied by ECIL. A 4-node prototype based on the MC 68020 processor (working at 16.67 MHz) was first established. This used the VME bus for communication. The VME backplane is 'natural' to the Motorola family of processors and was found to provide the necessary bandwidth and operational flexibility. Later, an 8-node prototype, based on the MC 68030 processor (working at 25 MHz) was developed. This 8-node cluster forms the backbone of the PACE architecture. The 128-node prototype is based on the MC 68030 processor working at 33 MHz. In order to enhance the floating point speed, ANURAG has developed its own custom floating point processor, ANUCO. The processor board has been especially designed to accomodate the MC 68881, MC 68882 or ANUCO floating point accelerators.

Subsequently, ANURAG configured a 2-node version of PACE, based on the Intel i860. Since the specific CPU boards used were not configured on a standard bus, the communications were established through a common multiported memory. This was done as a demonstration to prove that the concept was portable. Recently, the PACE architecture was sported on the SPARC II processor working at 40 MHz. The hardware was procured from Themes (France) through a Bangalore-based company called UBM.

System Architecture

PACE consists of a Front-End-Processor (FEP) connected to four super-clusters by means of VME-to-VME communication links. These VME-to-VME links provide high speed parallel (32 bit wide) communications between the two VME backplanes. The super-clusters are completely connected to VME-to-VME links. Each super-cluster has two CPUs exclusively devoted to communications. One CPU handles intra-super-cluster messages, while the other handles inter-super-cluster messages.

Each super-cluster has four clusters connected to it. Each cluster has eight CPUs connected on a VME backplane. The clusters are linked to the super-clusters by VME-to-VME links. Thus, each super-cluster has 32 CPUs, and four super-clusters can accommodate 128 CPUs. The CPUs within the cluster are completely connected over the VME bus. They communicate with each other by directly writing the messages into the appropriate buffer space over the VME bus. For communication across clusters within the same super-cluster, the CPUs within a cluster pass the message on to the super-cluster, which then passes on the message to the node in the destination cluster. Communication between nodes in different super-

clusters takes three hops, which involves two super-clusters, the source cluster and the destination cluster.

The latest offering of PACE is called PACE+ and is based on the HyperSPARC node running at 66 MHz. The memory per node is expandable up to 256 MB. The basic hardware again comes from Themes, France.

Programming Environment

The programming environment of PACE is simple and straightforward. The user interacts with the front-end processor (FEP), which is a standard UNIX engine with HyperSPARC processor running Solaris. The parallel processor is treated as a resource of the front-end processor. The user writes his program in a sequential fashion (this is called the 'host' program). All computationally intensive portions of the programs are written as subroutines, which are executed in parallel on the parallel processor. The user therefore needs to parallelize only the computationally intensive parts of the program, which are treated as subroutines (called the 'node' program) to be called by the host program.

In order to enable the user to create, debug and execute his programs, ANURAG has written a parallel programming environment called ANUPAM (ANURAG's Parallel Applications Manager). ANUPAM runs under UNIX and consists of several modules and utilities. These include Preprocessor, Simulator, Queue Manager, Run-Time Libraries, Communications Debugger, Source-Level Debugger, Parallel Library and other utilities. The ANUPAM software only depends on the availability of UNIX at the front-end. The software is portable across machines with very few modifications (the modifications relate to the physical addresses of the CPU boards).

Applications

Several application programs have been run on the various models of PACE. These include Linpack, FFT, neural networks simulation, FEM codes and several CFD codes. The PACE 128 system, based on the Motorola 68030 processor and MC 68882 co-processor, delivered over 30 Mflops speed for large problems. The speed per processor node was 0.33 Mflops. Later, this was enhanced to 0.75 Mflops per node incorporating ANURAG's custom FPU ANUCO. With the SPARC II processors, the speed is 4.5 Mflops per node. The latest SPARC processors will offer higher performance.

NAL's FLOSOLVER

Background and Evolution

The National Aerospace Laboratories (NAL), located at Bangalore, is a major national laboratory of the Council for Scientific and Industrial Research of the Government of India. In 1986, NAL started the project to design, develop and fabricate suitable parallel processing systems to solve fluid dynamical and aerodynamical problems. The project was motivated by the need for a powerful computer in the laboratory and was influenced by similar international deveopments. The existing UNIVAC system then existing at NAL could not meet these computational requirements.

The parallel computer of NAL is called FLOSOLVER, and was the first Indian parallel computer to become operational in 1986. Since then, a series of parallel computers have been built at NAL, which include the FLOSOLVER Mk1 and Mk1A, which were four-processor systems based on 16-bit Intel 8086/8087 processors, the FLOSOLVER Mk1B, an eight-processor system in this series, the FLOSOLVER Mk2, based on 32-bit Intel 80386/80387 processors, and the latest version, the FLOSOLVER Mk3, based on the RISC processor i860 from Intel.

System Architecture and Programming Environment

The present version of the FLOSOLVER Mk3 is based on eight i860 RISC processors with an on-board memory of 64 MB per processor. The system bus is Multibus II and has a bandwidth of 40 MB/sec. The communication between the processors is assisted through the message passing co-processor (MPC) and high speed direct memory access (DMA) controllers available on each of the boards.

Thus, it can be seen that the system architecture of NAL's FLOSOLVER is very similar to the architecture of BARC's ANUPAM single cluster. The major software development carried out for FLOSOLVER includes a simulator, which can run on any UNIX machine and can be used as a front-end. The parallel FORTRAN pre-processor has also been developed. The concurrent executive, a varient of Intel's iMRX, runs on each node. The standard C language and FORTRAN compilers are supported. A vectorizer is also available.

Applications

The application of NAL's FLOSOLVER is dominantly focussed on the weather forecasting code T80 under a project from the Department of Science and Technology of the Government of India. The FLOSOLVER has also been extensively used by the scientists of NAL to solve their computational fluid dynamics problems.

With the sustained speed of over 15 Mflops, NAL scientists have actively used FLOSOLVER for CFD problems. A panel code to compute aerodynamic coefficients on an

entire aircraft, with over 6000 panels, has been completed in less than an hour. Direct numerical simulation of the initial evolution of a turbulent axisymmetric wake and a 3D Euler computation of flow past a wing have been done on this machine. In one of the latest applications, Global Circulation Model code for weather forecasting running on a Cray has been sucessfully parallelized on FLOSOLVER.

C-DOT's CHIPPS

Background and Evolution

The Centre for Development of Telematics (C-DOT) was launched by the Government of India as a mission project to develop indigenous digital switching technology. C-DOT completed its First Mission in 1989 by delivering technologies of RAX for Rural Exchanges, and MACS for secondary switching areas. In Febraury 1988, a development contract was signed by the Department of Science and Technology and C-DOT, under which C-DOT was to design and build a 640 Mflops 1000 MIPS peak parallel computer. C-DOT set a target of 200 Mflops for sustained performance.

System Architecture

The CHIPPS, C-DOT's High Performance Parallel Processing System, is based on the single algorithm multiple data architecture. Drawing advantages of both SIMD and MIMD, the architecture provides coarse grain parallelism with barrier synchronization. It also provides uniform start-up and simultaneous data distribution across all configurations. It is realized using off-the-shelf hardware and software technology tools. The CHIPPS is designed to support large, medium and small applications. The range includes a large 192-node machine, a 64-node machine and a compact 16-node machine.

A flexible Interconnection Network (ICN) configured by the Main Controller (MC) interconnects Processing Elements (PEs) and global memory (MDM) banks. While raw data for computation is written by the MC into the memory mapped banks, the program is broadcast to the PEs through high speed serial links. The computed results are transferred simultaneously to the banks, which are then stored onto the disks. The MC broadcasts the commands to the PEs and MDMs and synchronizes all operations at task levels.

Programming Environment

The CHIPPS provides Data Parallelism as its program paradigm. Application programs written for sequential machines can be easily parallelized by identifying the tasks which are data-independent. The application program is functionally divided as tasks, viz. I/O, data

management, computation. The computation is performed at PE and the control and I/O at MC.

The user interface is provided through the system library to perform tasks such as data transfer, switch setup, execution control, etc. The system library, a collection of high level language callable routines, coordinates with the kernels at the PE and the MDMs, and thus helps the user to parallelize the application program in the UNIX environment at the MC. A debugger at the MC, which allows setting of break and watch points, viewing the values of selected variables etc., is an additional feature for the application program development.

Applications

The CHIPPS was originally designed primarily for weather forecasting and radio astronomy applications. Recently, several scientific kernel codes were ported onto the machine to demonstrate its general applicability in scientific and engineering applications.

Relative Performance of Indian Parallel Computers

In terms of speed, the Indian high performance computers may not be comparable to the best machines, such as the IBM SP-2 and SG Challenger, but the Indian machines may turn out cheaper compared to machines with lower power, for a similar performance range. For example, CDAC's PARAM 9000/SS model with Super Sparc has a peak performance of 0.96 Gflops for a 16-node system, whereas SGI Power Challenge has a peak performance of 5.76 Gflops for 16 processors, while the IBM Power 2 model 590 has a peak performance of 4.22 Gflops for 16 processors. The future generation teraflop machines of CDAC based on DEC Alpha processors are supposed to be a match to the performance of the best parallel machines available. The projected performance of the future CDAC high performance computers is hown in Fig 2.

More details on the applications and architectures of the machines discussed above may be found in [8].

A Glimpse of Certain Indian Research Efforts

The five Indian Institutes of Technology (IITs) located at Bombay, Madras, Kharagpur, New Delhi and Kanpur, the Indian Institute of Science (IISc) located at Bangalore, the Centre for Mathematical Modelling and Computer Simulation (CMMACS) located at Bangalore, and a number of other organizations, have made several significant research contributions to high performance computing. Notable among them is the novel architecture based on an hierarchical network of hypercubes [1], parallelizing diverse applications in the areas of medical imaging [2], VLSI layout [3], logic programming [4], scientific computa-

tion [5], multiprocessor operating systems [6], and interconnection networks [7]. There have been several success stories of building prototype/experimental parallel machines in some of the above academic institutions.

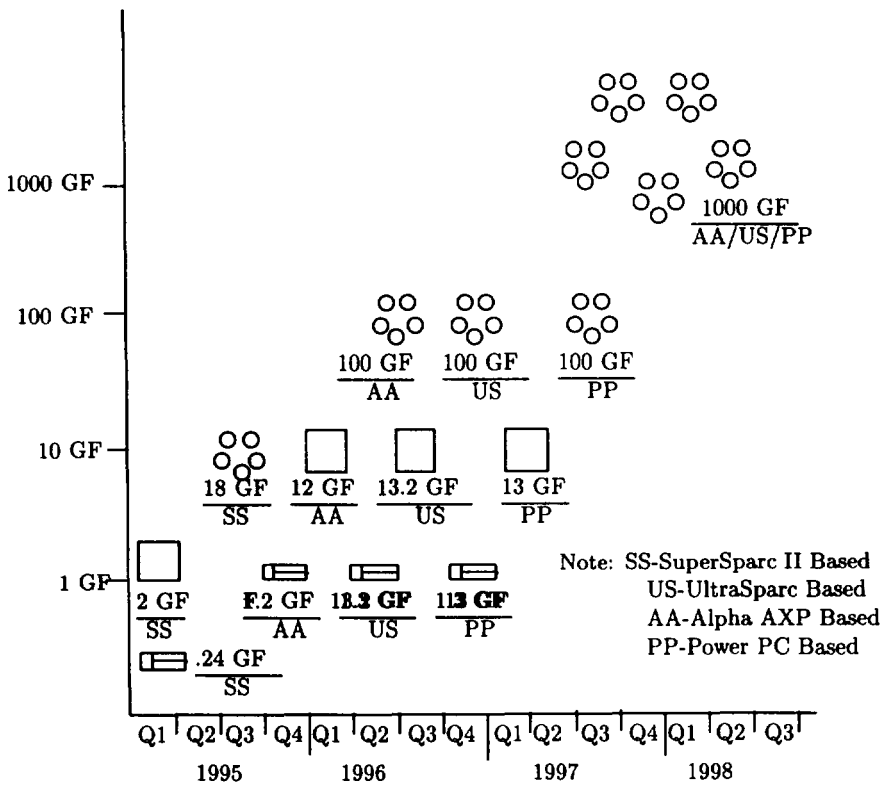


Figure 2: Roadmap to teraflops of PARAM 9000

Support for Indian High Performance Computing Activities

Typical high performance computing facilities in India include Cyber 992, CD4360, R6000/580 cluster, DEC 10000/620, CDAC's PARAM, and Power Challenge at the Indian Institute of Science, Bangalore; Convex C220 at the Indian Institute of Technology, Kanpur; CDAC's PARAM at the Indian Institute of Technology, New Delhi; Cyber 180 at the Indian Institutes of Technology, Bombay and Kharagpur; and Convex 3820 at the Centre for Mathematical Modelling and Computer Simulation, Bangalore, and CRAY XMP/216 at the National Centre for Medium Range Weather Forecasting (NCMRWF) at New Delhi. Most of these computing systems have the support of PVM, and MPI to carry out extensive studies on parallel algorithms/architectures. Most of the parallel software development is carried out using Parallel C, Parallel Fortran and Occam on transputer-based systems.

At present, the major funding for high performance computing activities in India comes from government sources, such as the Department of Electronics, Department of Science and Technology, Department of Scientific and Industrial Research, in the form of sponsored research projects. Realising the significance of this activity, the Department of Electronics of the Government of India took all initiatives to set up the Centre for the Development of Advanced Computing (CDAC), in a mission mode. Several Indian and multinational firms, such as Tata Elxsi (India) Ltd., Tata Information Systems Limited, Motorola India Electronics Pvt. Ltd., Digital Equipment (India) Ltd. and Silicon Graphics, have initiated strong marketing and software development activities in India in recent months. Diverse applications drawn from scientific computing, particularly in the areas of signal processing, transaction processing, multimedia, virtual reality, and simulation are of interest to several of these private industries. The Indian software houses at present do not significantly contribute to parallel software development efforts, but the scenario may change in future. The government funded organizations primarily look at applications in the areas of weather modelling, aerodynamic simulation, finite element analysis. Some other applications are in the areas of analysis of aerospace, automotive and offshore structures; molecular modelling for drugs, pesticides and biotechnology, weather prediction, pollution studies and turbulence. Academic institutions, such as the Indian Institutes of Technology (IITs), the Indian Institute of Science (IISc), the Birla Institute of Technology and Science (BITS) and several regional and university engineering colleges, offer courses at the introductory and advanced level in the area of parallel processing. Most of these institutions have excellent research programmes in the areas of parallel architecture, parallel algorithms, compilers and operating systems for parallel computers and mapping diverse aplications to parallel machines. Future applications will broadly address the above areas, with a possible emphasis by private industries on commercial exploitation of high performance computing. The future government funding in this area may not increase significantly because the present emphasis is more on industrial collaborative efforts. Industrial participation and support, particularly from the multinational firms, is anticipated for an active promotion of this area. The general feeling is that most of the present applications are of interest to academic institutions and government industries. Unless sufficient applications of high performance computing are demonstrated in terms of their commercial viability, the future funding to support this important activity may be adversely affected in India.

A Brief Overview of High Performance Computing Activities in Other Countries of the Far East Region

The National Supercomputing Research Centre (NSRC) is the focus of high performance computing activities in Singapore. The emphasis in Singapore is to use high performance computing to promote innovation and industrial development within the country, and help its industry adopt advanced computing technologies, rather than promote more basic

research. The NEC SX-3 and IBM SP2 are the major facilities at the NSRC. The NSRC plans to procure a CRAY T90 by the end of 1995. Research focus ranges from traditional scientific areas to defence science, weather prediction and commercial applications. The NSRC in Singapore is funded by the National Science and Technology Board. Silicon Graphics and NSRC signed a memorandum in August 1994 to promote visualization technology, by establishing a Supercomputing Visualization Laboratory at the NSRC. The result of this memorandum is the supply of a pair of R8000 CPUs, two Indy workstations, an Onyx workstation, and a variety of software by SGI. In addition, there is a new programme in Computational Science at the National University of Singapore. Work is also carried out at the Institute of Systems Science of the National University of Singapore, and Nanyang Technological University, Singapore. Some industrial collaborative applications being investigated in Singapore are molecular simulation, scientific visualization, integrated land use and transportation modelling, financial modelling, and forecasting. The National University of Singapore currently operates a Cray J916 Supercomputer in addition to a few SGI and Convex machines. The Nanyang Technological University has an IBM SP2. The National Centre for High Performance Computing (NCHC) in Taiwan is equipped with an IBM ES-9000/860 (5 CPU), a Convex C3840 (4 CPU), an IBM SP1 (16 nodes), a Convex SPP (8 nodes), a Convex Meta (8 nodes), and handles projects in the areas of scientific databases and network applications. The activities are primarily supported by academic institutions, and to some extent by industry and NCHC.

The Tsinghua University in China concentrates on climate modelling, ocean circulation, turbulent flow, vision, and cognition as the main applications of high performance computing. In China, Galaxy 1 and IBM 3084 (2 processors) are being used for seismic processing at CNPC; Fujitsu M150, Galaxy 2 (4 processors), CRAY YMP (2 processors) for weather prediction at the National Weather Bureau (Beijing); and IBM ES 9000 for financial applications at the Commercial Bank (Shanghai). Other than universities, the Chinese government plans to promote this activity through projects, such as the National Natural Science Foundation.

Some of the high performance computers used in Australia are: CRAY YMP EL (Moldflow and BHP in Sydney), Convex 300 (Biochemical Research, Melbourne), CRAY YMP 464 (CSIRO, Melbourne), Intel Paragon and iPSC/860 (University of Melbourne), Fujitsu VP2000 and CM-5 (Australian National University). These systems are being used in the areas of image recognition and analysis, molecular modelling, visualization, climate research, polymers, tomography, geophysical and chemical applications. The support for such activities in Australia is through universities and research schemes, such as ARC and CRC (Collaboration Research Centres).

The Korean high performance computing systems in use are primarily the CRAY Y-MP systems, and application software in the areas of weather forecasting, computational fluid dynamics, computational chemistry, genetic engineering, and nuclear power plant safety analysis, have been developed. Some interesting research and development projects in

Korea are the KAICUBE project developed by KAIST (Korean Advanced Institute of Science and Technology) in 1993 and the TICOM IV Project sponsored by the Ministry of Communication and Science and Technology. An 8 CPU (i860) machine KAICUBE - 860/8 with a 320 Mflops rating – is the target of the KAICUBE Project, whereas the TICOM IV Project (1994–1998) aims at 256 processors (P6) with 20 GIPS performance, 1 GB/node memory for large online transaction processing applications.

In New Zealand, DEC Alpha 2000/300 AXP (University of Waikato), Silicon Graphics Iris Indigo workstations and Sparc 10 dual processor (Victoria University), DEC Station 5000/200 (Lincoln University) and other high performance computers are being used for application software development in the areas of finite element analysis, image processing, seismology, combustion modelling, weather forecasting, and computational chemistry applications.

In Hong Kong, the University of Science and Technology is equipped with an Intel Paragon with 140 processors (75 Mflops peak performance per processor) and 5 GB memory and 35 GB parallel disk array; and an SGI Onyx parallel processor with an 8 processor system (MIPS R4400 chip), 512 MB memory (shared), 10 GB disk memory. The Chinese University of Hong Kong has a DEC mpp 12000 massively parallel computer with 8192 processors, with Maspar parallel C, parallel FORTRAN, and parallel math library. The computing facilities at the Hong Kong University and City Polytechnic of Hong Kong are configured around an IBM SP1 with 8 processors and a CRAY YMP8/86A with 8 processors respectively. Some of the interesting applications studied in the various universities of Hong Kong include, convection induced turbulence simulation, parallelizing large linear programming problems, application of two level finite element method, Monte Carlo and molecular dynamics studies, and parallel implementation of neural networks. Most of the high performance activity in Hong Kong is supported in universities.

Some Future Predictions

One of the toughest things is to predict the future, more so for high performance computing. In future, the Network Of Workstations (NOW) will be an attractive alternative to parallel machines, as far as high performance computing solutions are concerned. Such an alternative will be more realistic in future, with fast advances in communications technology, particularly in the areas of high speed switching and ATM networks. But the interest in massively parallel machines and heterogeneous computers will continue for teraflop range applications. The support for such highly specialized applications may have to come through government sources, since such applications will be from the research community, rather than from industry. The trend may be to network several parallel machines (even of different architecture) through high speed networks. Future parallel systems will be networks of heterogeneous computers, comprising some of the following: workstations, PCs, shared-memory multiprocessors, and special-purpose machines. There will be greater

integration of parallel computation, high-performance networking, and multimedia systems. The popular parallel programming languages will be based on C, C++ or some object-oriented paradigm.

Standard parallel languages must be developed if parallel computers are to achieve the same level of popularity as that of sequential machines. Future compilers or operating systems will take charge of distributing parallelism onto different processors and also of exploiting levels of parallelism in an application. There will be more attempts to run databases such as Oracle, Informix, on parallel machines. Other than business applications, recreation too will drive parallel computing.

Some "functional" versions of C and Fortran will facilitate parallelizing compilers. Tools will also play a very important role in developing parallel applications, making them portable across different architectures. Coarser grained objects than we have today may emerge. Multiple objects will work in parallel to solve a problem. These objects will use the parallel processing constructs.

In the next decade or so, machines in the commercial arena will be shared-memory multiprocessors. Hopefully, we will have multiple nodes with independent memory virtualized as a shared-memory system by the operating system. A multiparadigm, portable, standard substrate is essential if parallel computers are ever to flourish.

The massively parallel machines will continue to be used for the same type of applications as broadly classified under the HPCC (High Performance Computation and Communication) project in the USA. However, there will be more interest in the use of high performance computing techniques for commercial applications, e.g. on-line transaction processing, multimedia, etc. High speed networking will enable users to access teraflop machines across continents. Applications involving image processing, virtual reality, and simulation with possible emphasis on defense strategies will assume more significance. Data-intensive business applications, such as videoconferencing, advanced graphics and multimedia, will take advantage of parallelism. To sum up, high performance computing may not be projected to be a hype any more!

Acknowledgement

Dr. Vijay P. Bhatkar, Executive Director, Centre for Development of Advanced Computing, Pune, India is profusely thanked for providing details of C-DAC's parallel machines through his technical report. Dr. K. Neelakantan, ANURAG, Mr. M. Periasamy, C-DOT, and Dr. U. N. Sinha, NAL, are also acknowledged for providing details about the parallel machines in their organisations.

References

[1] J. Mohan Kumar and L.M. Patnaik: *"Extended Hypercube: A Hierarchical Interconnection Network of Hypercubes"*, **IEEE Transactions on Parallel and Distributed Systems**, Vol.3, No.1, Jan. 1992, pp.45-57.

[2] K. Rajan, L.M. Patnaik and J. Ramakrishna: *"High-Speed Computation of the EM Algorithm for PET Image Reconstruction"*, **IEEE Transactions on Nuclear Science,** Vol.41, No.5, October 1994, pp.1721-1728.

[3] B.B. Prahlad Rao, L.M. Patnaik and R.C. Hansdah: *"Parallel Genetic Algorithm for Channel Routing"*, **Proceedings of the Third Great Lakes Symposium on VLSI, Kalamazoo,** Michigan, USA, March 1993, pp. 69-70.

[4] A.V.S. Sastry and L.M. Patnaik: *"OR-Parallel Evaluation of Logic Programs on a Multi-Ring Dataflow Machine"*, **New Generation Computing,** Vol.10, No.1, 1991, pp.23-54.

[5] P.C. Mathias and L.M. Patnaik: *"Systolic Evaluation of Polynomial Expressions,*, **"IEEE Transactions on Computers**, Vol.39, No.5, May 1990, pp.653-665.

[6] Pradeep Hatkanagalekar: Concurrency Issues in Shared-Memory Multiprocessor Operating System Kernels, Ph.D Thesis, Department of Computer Science and Engineering, Indian Institute of Technology, Bombay, 1995.

[7] Ravi Mittal, B. N. Jain and R. K. Patney: *"Link Augmented Binary (LAB)-Tree Architecture"*, **IEEE Proceeedings E: Computer and Digital Techniques**, Vol. 140, No.2, March 1993, pp. 127-133.

[8] Vijay P. Bhatkar: Parallel Computers Galore in India, Technical Report, Centre for Development of Advanced Computing, Pune, India.

IV. Parallel Computing: A Latin American Perspective

by

S. W. Song

Introduction

High-performance computing has almost always been associated with the most expressive new results obtained in various areas of science and technology. The appearance of the so-called supercomputers has opened new horizons for scientific research and development, enabling observations and experiments considered impossible or economically unfeasible until then.

High-performance computing is intimately related to the solution of the so-called "Grand Challenges" that include such problems as forecasting of weather and climate, material sciences, structural biology, chromodynamics, transport, ocean sciences, medicine and health.

According to the recent (10 November 1995) TOP500 Supercomputer Sites report [12], the topmost 500 supercomputers are distributed as follows:

Table 1

Continent or country	No. of supercomputers in TOP500 list	Percentage
USA	269	54
Europe	139	28
Japan	73	14
Others	19	4
Total	500	100

Notice the relatively small number (4 per cent) of supercomputer sites located outside the USA, Europe and Japan. Among these 19 supercomputer sites, only two are located in Latin America, one in Mexico and one in Brazil. At the Universidad Autónoma Metropolitana, Istapalapa, Mexico, the Silicon Graphics Power Challenge occupies the 268th position among the TOP500 list. In Brazil's INPE (Instituto Nacional de Pesquisas Espaciais –

National Institute of Space Research), the Nippon Electric Company SX-3/12R occupies the 420th position. This machine is used in weather forecasting research. The following table gives a summary.

Table 2

Country	Institution	Equipment	Usage	Ranking
Mexico	Univ. Aut. Metropolitana	SGI Power Challenge	Academic	268
Brazil	INPE/CPTEC	NEC SX-3/12R	Weather	420

(Source: TOP500 Supercomputer Sites – 10 November 1995, Report [12])

The number of supercomputer sites located in Latin America was a little larger two years ago. According to the 1993 TOP500 Supercomputer Sites report, there were two supercomputers located in Brazil and another two in Mexico, as shown in the following table.

Table 3

Country	Institution	Equipment	Usage	Ranking
Brazil	INPE/CPTEC	NEC SX-3/12R	Weather	93
Mexico	Univ. Nac. Aut. de Mexico	Cray Y-MP1/132	Academic	292
Mexico	ITESM	IBM 9076-001 SP-1	Academic	408
Brazil	UFRGS	Cray Y-MP2E/232	Academic	461

(Source: TOP500 Supercomputer Sites – 11 November 1993, Report [11])

Evolution of the TOP500 equipment, in terms of architecture, is of special interest. Notice the gradual and steady increase of the MPP (massively parallel processor) technology during the past three years (see the following table [12]). It is quite certain that parallel computing should play a very important role in supercomputing in the next decade.

Table 4

Month/year	No. of MPP systems in TOP500 list	Percentage
Jun/1993	156	31
Nov/1993	187	37
Jun/1994	227	45
Nov/1994	239	48
Jun/1995	230	46
Nov/1995	284	57

With respect to CPU technology, off-the-shelf CMOS seem to gain ground at an extremely fast pace. The following is based on the TOP500 supercomputer sites during the past three years:

Table 5

Month/year	No. of systems using off-the-shelf CMOS	Percentage
June/1993	109	22
Nov/1993	124	25
Jun/1994	193	39
Nov/1994	242	48
June/1995	322	64
Nov/1995	364	73

Supercomputing Systems in Latin America

National Supercomputer Centres

As shown , the number of supercomputers among the TOP500 list located in Latin America is very small. However, there are a considerable number of supercomputers outside the TOP500 list. A very important role in this dissemination of high-performance computing is played by the governments. In Brazil, the Ministry of Science and Technology created the SINAPAD programme (Sistema de Centros Nacionais de Processamento de Alto Desempenho – National High-Performance Processing Centres). This is similar to the National Science Foundation Supercomputer Centers in the United States, however, with a much smaller budget. The mission of SINAPAD is to provide modern computing services of quality and high capacity to researchers and professionals, as a means to diffuse high-performance computing to the various segments of science and technology. Five such national supercomputer centres (called CENAPAD— Centro Nacional de Processamento de Alto Desempenho) have been installed or planned, in the states of Rio Grande do Sul, São Paulo, Rio de Janeiro, Minas Gerais and Ceará. One characteristic is that these centres, once installed, are supposed to be self-supporting through the services they provide. The central coordination of the CENAPAD is established by the Ministry of Science and Technology, through its FINEP (Financiadora de Estudos e Projetos – Research and Project Funding Agency).

The SINAPAD programme is not limited to the national supercomputer centres, which aim to diffuse the usage of high-performance technology. To a lesser extent, there is also a programme to finance the research and design of high-performance hardware. A much smaller budget is dedicated to this end. The research and development groups deemed to be suitable for this high-performance hardware design programme are the Universidade Fed-

eral do Rio de Janeiro, through its COPPE (Coordenação dos Programas de Pós-Graduacao de Engenharia) and the Universidade de São Paulo, through its LSI (Laboratório de Sistemas Integráveis).

In the following we give a brief description of three of the five CENAPADs:

CESUP – RS

CESUP – RS (Centro Nacional de Supercomputação na Região Sul) was the first national supercomputer centre to be inaugurated (June 1992). It is installed at the Universidade Federal do Rio Grande do Sul.

The hardware configuration of CESUP – RS is as follows:

Cray Research Y-MP 2E/232 with two central units, each with:

- peak performance of 330 Mflops
- memory of 256 Mbytes
- disk space of 16 Gbytes

CENAPAD – SP

CENAPAD – SP (Centro Nacional de Processamento de Alto Desempenho de São Paulo), the second national centre (inaugurated in March 1994) is located at the Universidade Estadual de Campinas. Its hardware configuration is as follows:

IBM 9076 SP1 – 8 processors, each with:

- memory of 256 Mbytes
- disk space of 2 Gbytes

IBM ES 9021 – model 711

- peak performance of 563 Mflops
- memory of 512 Mbytes
- extended memory of 512 Mbytes

CENAPAD – RJ

CENAPAD – RJ (Centro Nacional de Processamento de Alto Desempenho do Rio de Janeiro) is installed at LNCC (Laboratório Nacional de Computação Científica). The hardware configuration is as follows.

IBM SP-2 with 16 processors, each with:

- memory of 256 Mbytes
- local disk of 2 Gbytes

- 2 data servers RS/6000 mod. 980 with 55 Gbytes
- total peak performance of 2 Gflops.

A planned upgrade (February 1996) will elevate the processing capacity to 8.7 Gflops.

Other supercomputer systems

A partial list of other supercomputer systems is included in Appendix A.

Computing in Latin America

As in the case of many third world countries, exports of products are essential to the national economy. Some Latin American countries have shown success in their capacity to develop and export software. As reported in [3], exports of software developed in Chile amounted to approximately US$ 22 million in 1993. In addition to application packages and software utilities, the exported products include applications dealing with mining and forestry, in which Chile has special experience.

In 1992, the Brazilian Ministry of Science and Technology, through CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), launched an aggressive nation-wide programme, called SOFTEX 2000. What follows is based on [20], which can be consulted for more details. The SOFTEX 2000 programme is supported by the United Nations Development Programme (UNDP). The main goal is to redirect and steer the Brazilian industry towards the development of software of quality for export. Brazil presents an active and high-potential economy, with the 8th Gross National Product (GNP) in the world. Many factors show the strong vocation for informatics: Brazil hosts a large number of national and international software companies, with professionals trained through many high-quality university courses in computer science, a nationwide computer network (RNP – Rede Nacional de Pesquisa) interconnecting virtually all the major universities and research institutes. Expenditure on informatics in Brazil, as compared to the GNP, presents similar indices to those observed in more developed economies, such as France, Germany and Japan. One notable area is the usage of informatics and networking in transactional banking systems, which is certainly amongst the most advanced in the world. Research by Andersen Consulting shows that one in each three homes of classes A and B has a computer. The total of computer equipment is 1.24 million, about 42 per cent of which with fax/modem. The following table contains some interesting comparative data:

Table 6: (In billions of US dollars)

Country	Hardware	Software products	Services in Informatics	Total	Total/GNP	Total per capita
USA	85.6	31.8	58.8	176.2	2.3	646
Japan	35.5	6.4	30.2	72.1	2.04	295
Germany	17.0	5.8	15.0	37.8	1.94	465
France	10.4	4.1	12.0	26.5	2.00	466
UK	11.7	4.2	10.2	26.1	2.41	438
Italy	6.2	3.3	7.1	16.6	1.35	292
Brazil	5.2	0.9	2.7	8.9	2.00	63
Netherlands	3.4	1.7	2.8	7.9	2.36	499

(Sources: Secretaria de Informática, Brazil, also [13])

Since the end of the Informatics Reserve Market Policy in 1990, a considerable effort has been expended towards a commercial integration into international markets. (Refer to [17 and 19] for some literature on the Brazilian national policy on informatics.) The SOFTEX 2000 programme, a joint initiative between the government and private industry, aims to foster the export of high quality software as a strategic economic alternative by the end of this century. It is viewed as a high priority programme by the government to promote software in the international market. An outpost office has already been established in Florida, USA, as a reference point of the programme to facilitate access and contact with the North American market. More specifically, the goals of SOFTEX 2000 are to conquer one per cent of the international software market by the end of this century, to capacitate more than a thousand companies and generate 50,000 new skilled jobs. This ambitious programme is coordinated by a special team from CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico). It is the role of this central coordination to formulate the strategy and allocate resources of CNPq and the United Nations (UNDP). In addition to the central coordination, there are regional entities to promote the software segment in their respective regions of influence.

Among the small Latin American countries (defined as having less than ten million inhabitants), some have shown success in serving as hosts for foreign investment and as export platforms for international corporations. These corporations take advantage of cheap labour to set up data processing centres in the Dominican Republic, Jamaica and Costa Rica [7]. The latter is host to several major international corporations to be a regional leader in furnishing international networking to Central America.

Research in Parallel Computing

A partial list of universities in which there is research on parallel computing follows. The mentioned areas are for illustrative purposes.

Table 7

University	Examples of Research Areas
Univ. Federal de Pernambuco, Brazil	Parallel Architechtures Parallel Algorithms
Univ. de Brasília, Brazil	Parallel Architectures Concurrent Programming Distr. Operating Systems
Univ. Fed. Mato Grosso Sul, Brazil	Parallel Algorithms
Univ. Fed. Minas Gerais, Brazil	Parallel Processing Parallel Algorithms
Univ. de São Paulo (Eng.), Brazil	Parallel Computer Architectures Prallel Languages and Compilers
Univ. de São Paulo (Comp. Sci.), Brazil	Parallel Algorithms Systolic Algorithms
Univ. Estadual Campinas, Brazil	Parallel Algorithms Efficient Implementation Issues
Univ. Fed. São Carlos, Brazil	Parallel Algorithms
Pont. Univ. Católica Rio de Janeiro, Brazil	Parallel Algorithms
Univ. Fed. Rio de Janeiro, Brazil	Parallel Computer Architectures Parallel Algorithms Superscaler Architectures
Univ. Fed. Rio Grande Sul, Brazil	Parallel Architectures
Centro de Calculo, Fac. de Ingenieria, Uruguay	Parallel Processing on Workstations
Inst. Tec. Autónomo de México, Mexico	Distributed Systems
Universidad de Chile, Chile	Parallel Algorithms Cellular Automata
Univ. Simón Bolívar, Venezuela	Parallel Algorithms Parallel Programming Tools

In Brazil the following universities have official graduate programmes (recognized by the Ministry of Education) in computer science:

Pontificie Universidade Católica do Rio de Janeiro,

Universidade de São Paulo (campus São Paulo and São Carlos,)

Universidade Estadual de Campinas,

Universidade Federal do Rio de Janeiro,

Universidade Federal de Pernambuco,

Universidade Federal de Minas Gerais,

Universidade Federal do Rio Grande do Sul,

Instituto Nacional de Pesquisas Espaciais,

Universidade Federal da Paraíba,

Universidade Federal do Paraná,

Universidade Federal de Santa Catarina,

Universidade Federal de São Carlos,

Universidade de Brasilia,

Instituto Militar de Engenharia.

Among the above list, the first eight also have Ph.D. programmes and most of the graduate programmes include the area of parallel computing.

In [14] a very careful study of the the various aspects of computer science research in Mexico is presented. It contains also some interesting statistics concerning the important question of human resources in computer science of various Latin American countries. Brazil has the largest number of around 380 Ph.D.s in computer science (with one Ph.D. for every 433 thousand inhabitants), Chile has 70 (one Ph.D. for every 171 thousand inhabitants), Mexico has only one Ph.D. in computer science per million inhabitants. In Mexico, the Ph.D.s in computer science are concentrated in one public and five private centres. Of the latter, four are departments of two private universities: Instituto Autónomo de México (ITAM) and Instituto Tecnologico de Estudios Superiores de Monterrey (ITESM). The other private institution is the Laboratorio Nacional de Informatica Avanzada (LANIA). The Sección de Computación of the Department of Electrical Engineering of the Instituto Politecnico Nacional is the only public centre that contains a group of Ph.D. researchers in computer science.

Conferences and Journals in Computer Science

Most countries maintain their local computer societies, such as the Sociedade Brasileira de Computção (SBC), Sociedad Chilena de Ciencia de la Computación (SCCC), Sociedad Argentina de Informática e Investigación Operativa (SADIO). There is also a Latin American Computer Society – CLEI – Centro Latino Americano de Estudios en Informática.

The Brazilian SBC organizes a main annual Computer Science Conference (with an attendance of more than 2,000 at the August 1995 meeting), as well as about a dozen specific symposia in such areas as databases, software engineering, computer graphics and image processing, artificial intelligence, computer networks, conception of integrated circuits, neural networks, music and computers, informatics in education, programming languages, etc. Some of these symposia have international programme committees, with submissions from all around the world. Since 1987, SBC started the annual symposium on parallel processing, currently named SBAC-PAD – Simposio Brasileiro de Arquitetura de Computadores e Processamento de Alto Desempenho (Computer Architecture and High-

Performance Processing Symposium). The specific area of parallel computing has grown considerably during the past few years in Brazil. In last August 1995, the 7th SBAC-PAD was held in Canela, Rio Grande do Sul, with nearly 600 participants.

The Chilean SCCC organizes an annual conference called the International Conference of the Chilean Computer Science Society, a conference of high quality with an international programme committee.

In 1992, the Department of Computer Science of the Universidade de São Paulo (DCC/IME/USP) sponsored the international symposium LATIN'92 (*Latin American Theoretical INformatics*), the first of a series of symposia on theoretical computer science to be held in Latin America. The conference proceedings are published as Lecture Notes in *Computer Science* by Springer-Verlag. The second edition of LATIN was held in 1995 in Valparaíso, Chile. The third is planned to be held again in Brazil, at the Universidade Estadual de Campinas, in 1997.

In 1993, the First South American Workshop on String Processing was organized and held at the Universidade Federal de Minas Gerais. The second workshop was repeated in Chile, held immediately after LATIN, in April 1995. The third workshop will take place at the Universidade Federal de Pernambuco, in August 1996.

An annual Latin American conference is organized by CLEI, called Conferência Latinoamericana de Informática. This conference is held each year at a different Latin American country.

The Sociedade Brasileira de Computação (SBC) publishes the journal *Revista Brasileira de Computação*. In 1994, SBC inaugurated a new international journal, named *Journal of the Brazilian Computer Society* (JBCS), with an international editorial board. JBCS is published three time a year. The third issue, published in July 1995, was dedicated to parallel computing.

International Cooperation – A New Experience

The cooperation between the university and industry has been much enphasized, particularly in the European Community's ESPRIT programme. Cooperative programmes involving diverse partners from academia and industry have been adopted as the *sine qua non* condition in many recent international cooperation programmes.

In the area of parallel computing, the ITDC'94 programme (Information Technologies for Developing Countries) can be considered as extremely important to the development of this area in Latin American countries. ITDC was an initiative of the Commission of the European Communities (CEC). The purpose of this initiative is to provide research infrastructure, in terms of parallel computers, to aid research groups located in developing countries. The goal is to establish scientific cooperation between research institutions in developing countries and in Europe. The CEC funding, through ITDC, includes the

purchase of European parallel computers, as well as costs for installation, training and travel. The call for projects was responded to with great enthusiasm, with more than 250 projects submitted from research groups located all around the world. The high competition ensures the quality of the more than twenty selected projects to receive funding. In Latin America, we can mention this with the following partial list of institutions that were selected: ITESM of Mexico, Universidad de Venezuela, Universidad de Chile, Universidade de São Paulo (two projects – one from the Department of Computer Science and another from Electrical Engineering), Universidade Estadual de Campinas (Computer Science and Electrical Engineering) Universidade Federal do Rio de Janeiro, etc.

Another CEC cooperative programme involving multiple partners is the ALFA programme (America Latina Formacion Academica), with the purpose of encouraging scientific cooperation between European and Latin American institutions, mainly in the formation of human resources.

In 1995, CEC launched the INCO programme (International Cooperation with Third Countries – Part C) with a total funding of ECU 208.98 million through 1998. The proposal should be transnational, involving at least two partners from different Member States of the European Union and at least one partner from a developing country, with strong preference to projects with at least two partners from separate developing countries in the same region. In each annual call for proposals, high priority areas are identified. One such area is Applications of High-Performance Computing, Parallel Programming and Networking (HPCN) with the following objectives:

- raising the awareness of and promoting the use of HPCN in industry
- training HPCN applications developers and users in industry
- establish HPCN best practice in a user environment
- stimulating the development of the HPCN infrastructure
- undertaking HPCN research as identified by application requirements.

This four-year programme will undoubtly boost the already effervescent parallel computing research in Latin American countries to even higher levels.

One notable point to be noticed in recent funding of research projects is the presence of collaboration with industrial partners, with preference to application-oriented research topics that would result in substantial social benefits. To illustrate this point we mention the CEC/ITDC project being coordinated by this author (Universidade de São Paulo (IME/USP), Departamento de Ciência da Computação, Grupo de Computação Paralela e Distribuída). This project has the collaboration of GMD FIRST, Berlin, and Parsytec (who provided the PowerXplorer 16/32 Parallel Computing System installed at IME/USP).

One of the goals of this project is to develop application-oriented software for numerically intensive applications. The chosen application area is meteorological computations and

environmental analysis (monitoring the dispersion of pollutants). The research part of the project in environmental and atmospheric applications will be conducted by researchers at the Department of Applied Mathematics and Department of Atmospheric Sciences of USP. We will also seek collaboration with governmental agencies for environmental control in the city and State of São Paulo. The goal of the project is to improve the understanding of the meteorological control processes on air pollution in São Paulo and develop operational techniques to predict local atmospheric conditions around São Paulo. The Department of Atmospheric Sciences of USP implemented the Regional Atmospheric Modelling System (RAMS) in 1990, which will be used as the main atmospheric simulation tool. The smog analysis and monitoring system DYMOS, developed in Germany [18], will be used for the simulation of air pollution in São Paulo. Comparisons between the atmospheric components of both DYMOS and RAMS will be carried out. RAMS has been undergoing a significant rearrangement of the code in order to run on clusters of workstations and massive parallel machines. The parallel version of RAMS was available for users at the end of 1994 and will be ported to the Parsytec system. As part of the ongoing collaboration between GMD and IME/USP, a fine grain message passing version of the DYMOS model will be developed.

Numerical modelling of the local air circulation in São Paulo is an important technique in understanding the dispersion processes associated with air pollution. The basic role of the sea breeze and the complex orography on the initiation of precipitation has already been explored. However, most of the heavy pollution episodes in São Paulo occur during the dry season. Thus, atmospheric modelling will be applied to winter-type typical situations in São Paulo, considering realistic distributions of surface characteristics, such as high resolution topography (order of a kilometre), type of soil and vegetation, surface roughness, moisture and albedo. Apart from the process studies, an operational technique to forecast the local flow will be developed on a nested version of RAMS with a coarse grid resolution of 32 kms. and a fine grid of 2 kms. covering the metropolitan area of São Paulo.

Design of Scalable Algorithms

As a joint work of this author with F. Dehne (School of Computer Science, Carleton University, Ottawa, Canada), we have investigated the problem and issues of efficient parallel implementations. The following is extracted from an ongoing joint project.

The main problem in parallel computing is the well known "software bottleneck". Current commercial applications of parallel machines are still mainly restricted to trivially parallelizable problems where communication requirements are obviously low. On the other hand, there is a large body of literature on parallel algorithm design for many non-trivial problems. However, these results suffer from the fact that there is no agreed model of parallelism that is close enough to existing machines to allow for a reasonable prediction of the speed of an implementation. The problem is obvious for PRAM based parallel algorithms, but even network based parallel algorithms are often very problematic and the speed obtained when implementing such algorithms on a commercial multiprocessor is frequently

very disappointing. Fine-grained parallelism of the PRAM algorithms require a high and frequently unreasonable amount of processors, in addition to the need for tightly synchronized lock-step operations among the many processors. It is therefore imperative to design models and algorithms in such a way that the theoretical complexity analysis matches the timings observed in actual implementations.

For geometric problems, such as visibility, convex hull, voronoi diagram, union of rectangles, etc., which exhibit massive communication requirements, by using a coarse-grained model, Dehne [8,9] has recently achieved considerable progress towards this goal. For many architectures, such results are a considerable improvement over existing methods. This applies in particular to previous fine-grained algorithms, even if they are (fine-grained) optimal. For example, it is impossible for fine-grained mesh algorithms, even optimal ones, to yield optimal speedups for ratios $n/p > O(1)$ (n = number of processors, p = number of data items) by applying the usual simulation method (also called "virtual processors" in many multicomputer operating systems). For $n/p > O(1)$, our methods are considerably faster. These algorithms are simple and easy to implement. The constants in the time complexity analysis are small. Except for a small (sometimes fixed) number of communication rounds, all other computation requires no communication. For coarse-grained machines, this new method implies communication through few large messages rather than having many small messages. This is important for machines like the Intel iPSC, where each message creates a considerable overhead. Dehne and his collaborators have implemented and tested prototypes of some algorithms on a CM5 and iPSC/860, and obtained very fast running times which neatly match the theoretical analysis.

Dehne's coarse-grained model

We now present the main ideas of Dehne's scalable parallel computational model for coarse-grained distributed memory multicomputers.

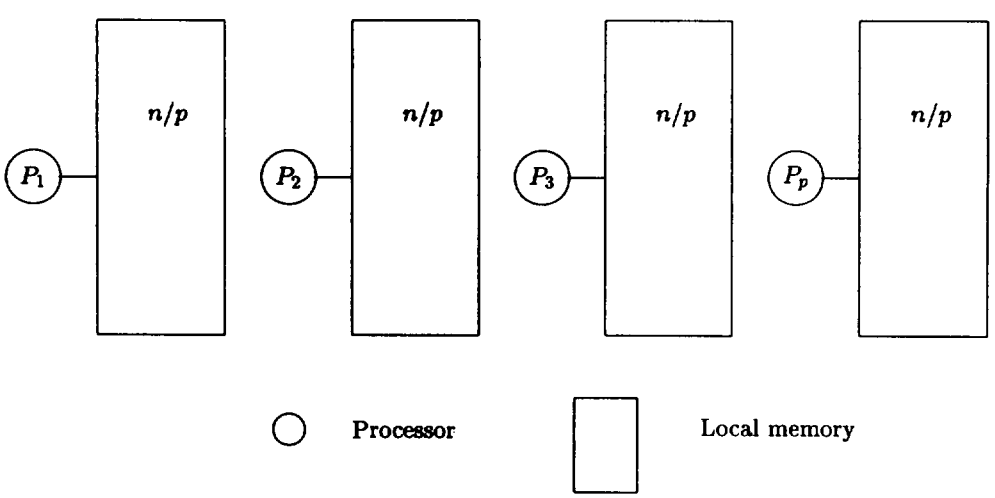


Figure 1: A distributed memory multiprocessor

Consider a problem of input size n and a parallel computer with p processors. Each processor thus gets n/p data items. PRAM algorithms often require $p = O(n)$. Thus the amount of data items to be placed in each processor is $n/p = O(1)$. Instead of this fined-grained model, we consider a more realistic case in which $p << n$, which is usually the case, even with current MPP (massively parallel processors). We want to design algorithms that are scalable, in the sense that they must be efficient for a wide range of n/p.

Consider Figure 1, where we have p processors $P_1, P_2, \ldots, P_p$, each with its local memory of size $O(n/p)$. The term coarse-grained is used here as to mean that the size of the local memory, $O(n/p)$, is "much larger" than $O(1)$. For instance, we may require $n/p \geq p$. These p processors are connected through some kind of interconnection network, e.g. mesh, hypercube, etc.

The basic idea is as follows: we first distribute the n input data items among the p processors, each with n/p. This is done by a partitioning scheme. The algorithm to be designed consists of a repetition of two phases or rounds: a computation phase followed by a communication round. In the computation phase, we attempt to use the best known sequential algorithms in each processor that processes their data independently. In the communication round, we shift data around, with each processor sending out a total n/p data items and receiving a total of n/p data items. We express this by the following generic scheme:

repeat the following k times

begin

computation phase
communication round

end

The goal is to design algorithms that require a small number of communication rounds, i.e. with the smallest possible k. In each computation phase we use the best possible sequential algorithm. In many geometric algorithms presented in [8 and 9], k is constant or $O(\log p)$. Since p is usually small as compared to n, the resulting algorithm will be highly efficient in practice. The studied geometric problems that can be solved as mentioned with k = constant or $O(\log p)$ include the following. Notice that efficient solution to some of these problems can be very important in computer graphics applications. The interested reader should refer to [8 and 9] for details.

1. Area of the union of rectangles
2. 3D-maxima
3. 2D-nearest neighbours of a point set
4. Lower envelope of line segments in the plane (also known as the visibility problem)
5. 2D-weighted dominance counting

6. Multisearch on balanced search trees, segment tree construction, and multiple segment tree search.

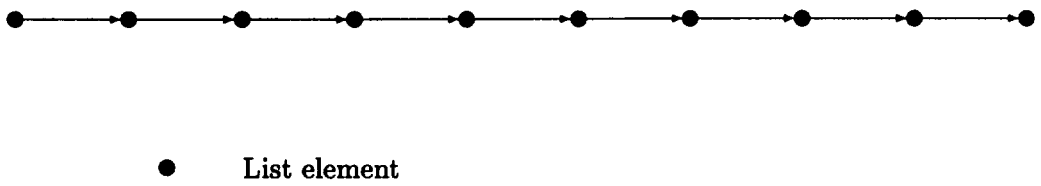


Figure 2: A linear linked list

In [10], we investigate the problem of list ranking. Consider a linked list of n elements (Fig. 2). The problem of list ranking consists of finding the location or distance of each element in a linked list with respect to the end of the list. It appears as subproblem in several graph and tree problems. Therefore an efficient solution for list ranking has direct consequences in other applications. A trivial sequential algorithm solves this problem in $O(n)$ time by traversing the list. Several PRAM list ranking algorithms have been proposed, with $O(\log n)$ time complexity.

In our coarse-grained distributed memory model, we distribute the input n in the p processors, as shown in Fig. 3.

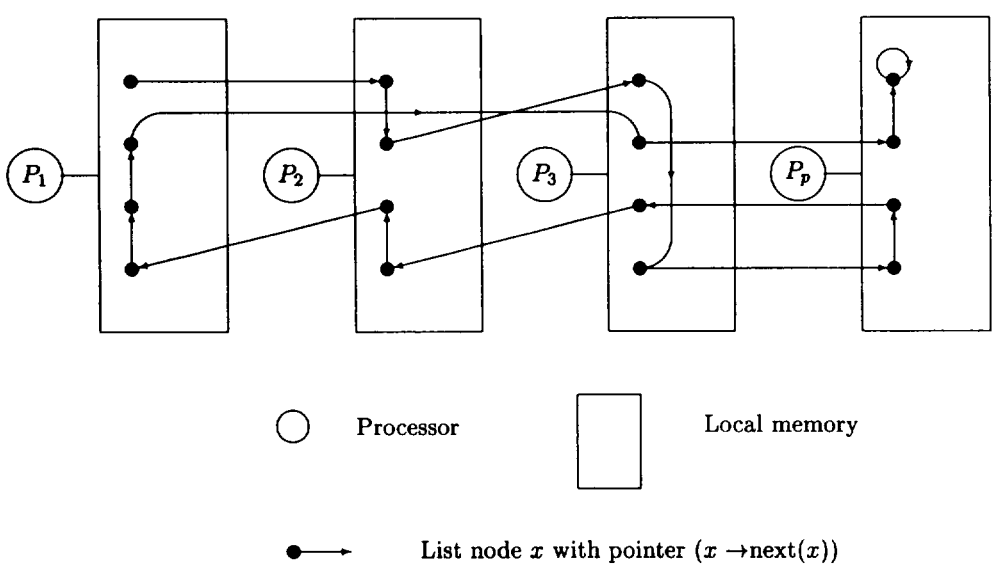


Figure 3: A linear linked list stored in a distributed memory multiprocessor

In [10] we present a scalable parallel algorithm for the n-element list ranking problem using p processors. Two versions of the algorithm are given. The first version requires, with high probability, log $(3p)$ + log ln (n) communication rounds. The second version requires only $O(r \log p)$ communication rounds, with high probability, where $r < \ln^*(n)$ is an extremely small number.

It is not our purpose to give the details here (which can be obtained from [10]). We wish to emphasize, however, the importance of this coarse-grained computing model for distributed memory systems. If we manage to design appropriate algorithms with a small number of communication rounds, then it is very likely that the resulting algorithm will be efficient in practice.

Virtual Shared Memory Systems

Shared memory systems have been popular due to many advantages and ease of their usage. Cache-coherent non-uniform memory access and non-coherent non-uniform memory access architectures have been proposed [1 and 16]. The hardware version of shared memory nevertheless incurs high cost of implementation. Implementation of shared memory in software, also known as virtual shared memory, constitutes a viable alternative [4 and 5].

Research in virtual shared memory systems are being carried out at the Universidade Federal de Rio de Janeiro on the NCP II machine (successor of the NCP I [[2]) and at the Universidade de São Paulo (IME/USP) [6 and 15].

The NCP II project is financed by the FINEP/MCT (Financiadora de Estudos e Projetos – Research and Project Funding Agency) of the Ministry of Science and Technology and started at the beginning of 1995. This constitutes one of the few projects supported by FINEP in the area of design and development of high-performance computers in Brazil. Another project financed by FINEP is being carried out at LSI/USP (Laboratório de Sistemas Integráveis, Universidade de São Paulo). The FINEP support amounts to roughly US$ 4 million.

The research of virtual shared memory at the Universidade de São Paulo (IME/USP) is supported by a joint GMD/CNPq international cooperation programme between Brazil and Germany. The counterpart in Germany is GMD FIRST, Berlin. One goal of the joint programme, among other goals, is to develop performance prediction models for parallel programmes, based on a virtual shared memory system (VOTE [6]) and runs on the GMD Manna machine. The following is an abstract extracted from [15].

The performance capabilities of virtually shared memory systems (VSM) still fails to seriously refute the impression that VSM seems to be a luxury, which provides conceptual simplicity at the expense of performance. The paper [15] is based on the VOTE VSM system running on top of a crossbar-based parallel computer system. The design of VOTE is a two layer approach, providing conceptual simplicity by means of a sequential consistency

at default and, moreover, providing a framework of functions to allow performance enhancements. This enables the programmer to use a customized programming style addressing any combination of sequential consistency's conceptual simplicity and the maximum performance achieved through message passing.

To validate the approach, performance overheads are discussed. The paper [15] shows the possibility of developing analytical performance models. Such models are able to predict the execution time for a generic number of processors and for different architectures. They can be used for bottleneck identification and, in some cases, serve as a guide for software improvement for both the application and the VSM system.

Conclusion

By observing the evolution of the TOP500 supercomputers in terms of architecture during the past few years, we notice an increasingly higher number of MPP (massively parallel processors) systems. We can be quite certain that parallel computing will play a crucial role in high-performance computing in the next decade.

Although there are very few supercomputers on the TOP500 list located in Latin America, we notice a considerable number of "smaller" supercomputer sites in Latin America, as shown in Appendix A.

The role of the federal government in establishing public scientific policy is very important. A notable example is the Brazilian Policy on Science and Technology of CNPq/MCT (Conselho Nacional de Desenvolvimento Científico and Tecnológico – National Science and Technology Development Agency) and FINEP/MCT (Financiadora de Estudos e Projetos – Research and Project Funding Agency), both of the Ministry of Science and Technology. These agencies determine the scientific and technological policies and provide the necessary funding. In addition to these two, there is still CAPES/MEC (Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) of the Ministry of Education that contributes to the formation of human resources by providing scholarships to graduate students to pursue their degrees, both in Brazil and abroad.

We examined the important role of funding resulting from international cooperation programmes. To obtain such fundings some of the usual requirements include the presence of international partners, including industrial partners. The projects are usually application-oriented and should produce measurable deliverables that benefit society. Although high-performance computing is costly, it can be crucial in solving some of the very pressing problems that can benefit mankind (environment analysis and planning, meteorology, etc.).

In more technical terms, we have examined the importance of defining new models of computation and the design of scalable parallel algorithms. Such algorithms should not only be efficient in theory, but principally in practical implementations. We also drew attention to the research of virtual shared memory that unites conceptual simplicity and maximum

Acknowledgment

This research was supported by FAPESP (Fundação de Amparo a Pesquisa do Estado de São Paulo) – Proc. No. 95/0767-0 and 95/6159-1, CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) – Proc. No. 523112/94-7 and PROTEM-2-TCPAC Proc. No. 680060/94-4, Commission of the European Communities through Project ITDC-207, and UNIDO (United Nations Industrial Development Organization). I would like to thank Professor Boleslaw K. Szymanski for inviting me to prepare this paper and for his comments. I am also grateful to Professor João Paulo Kitajima from the Universidade de Brasília who compiled some of the material used in this paper.

Appendix A

Supercomputer Systems

NEC SX3/12R

NEC SX3/12R (1 CPU, 2 pipes, 3.2 Gflops)

Locale: Instituto Nacional de Pesquisas Espaciais/Centro Meteorológico de Previsão e Tempo (INPE/CPTEC Weather Forecast Centre)

Observation: This equipment occupied the 186th position on the TOP500 Supercomputer Sites, Tennessee/Mannheim report of 9 November 1994. In the 11 November 1995 report, it occupies the 420th position on the TOP500 list.

Cray Research EL98 and J-90

Cray Research EL98 with 6 processors
Cray Research J-90 with 8 processors (to be upgraded to 16 processors)

Locale: Universidade de São Paulo (USP).

Cray Research Y-MP2E

Cray Research Y-MP2E – 2 CPUs.

Locale: Centro Nacional de Supercomputação, Universidade Federal do Rio Grande do Sul (CESUP/UFRGS)

Observation: one of the Brazilian National Supercomputer Centres (Centro Nacional de Processamento de Alto Desempenho – CENAPAD).

Cray Research EL94 and J-98

Cray Research EL94 and Cray J-98 (mid- 95)

Locale: Universidade Federal do Rio de Janeiro (COPPE/UFRJ)

IBM 9021-711

IBM 9021-711 with 1 VF.

Locale: Universidade Estadual de Campinas

Observation: one of the Brazilian National Supercomputer Centres (Centro Nacional de Processamento de Alto Desempenho – CENAPAD).

IBM 3090/600

IBM 3090/600-6VF (6 processors)

Locale: Petrobras, Rio de Janeiro.

IBM 9021-820

IBM 9021-820 (4 processors)

Locale: Petrobras, Rio de Janeiro.

IBM 3090

IBM 3090 with VF.

Locale: Universidade Federal de Santa Maria (UFSM)

IBM SP-1

IBM SP-1 (16 processors)

Locale: Laboratório Nacional de Computação Cientifica, Conselho Nacional de Desenvolvimento Científico e Tecnológico (LNCC/CNPq).

Observation: one of the Brazilian National Supercomputer Centres (Centro Nacional de Processamento de Alto Desempenho – CENAPAD).

IBM SP-1

IBM SP-1 (8 processors)

Locale: Universidade Estadual de Campinas (UNICAMP).

Observation: one of the Brazilian National Supercomputer Centres (Centro Nacional de Processamento de Alto Desempenho – CENAPAD).

IBM SP-2

IBM SP-2 (4 processors)

Locale: Universidade Federal do Ceará (UFCE)

Observation: one of the Brazilian National Supercomputer Centres (Centro Nacional de Processamento de Alto Desempenho – CENAPAD).

IBM SP-2

IBM SP-2 (4 processors)

Locale: FUNCEME/Ceará.

Observation: one of the Brazilian National Supercomputer Centres (Centro Nacional de Processamento de Alto Desempenho – CENAPAD).

Intel iPSC-860 / 8

Intel hypercube iPSC-860 / 8

Locale: Universidade Federal do Rio de Janeiro (COPPE/UFRJ)

Intel IPSC-2 / 16

Intel hypercube IPSC-2 / 16

Locale: Universidade de Brasília (Física/UnB).

Parsytec PowerXplorer

Parsytec PowerXplorer 16/32 (16 nodes each with a PowerPC 601 and T805 – 1.28 Gflops peak).

Locale: Universidade de São Paulo, Instituto de Matemática e Estatística (DCC/IME/USP).

Parsytec Multicluster/MP

Parsytec Multicluster/MP (Transputer).

Locale: Universidade Federal de Pernambuco (DI/UFPE).

Silicon Graphics Challenge

Silicon Graphics Challenge / 8 CPUs R4000

Locale: Universidade de São Paulo (LSI/USP).

Silicon Graphics Power Challenge

Silicon Graphics Power Challenge (6 processors)

Locale: Instituto de Estudos do Mar/Rio de Janeiro

Observation: This equipment occupied the 392th position on the TOP500 Supercomputer Sites of the TOP500 Tennessee/Mannheim report on 9 November 1994.

Silicon Graphics / 2 CPUs R800

Silicon Graphics / 2 CPUs R800

Locale: Centro Tecnológico Aeroespacial – Instituto de Estudos Avançados (CTA/IEAv)

Silicon Graphics Onyx

Silicon Graphics Onyx (4 processors – shared memory)

Locale: Centro Nacional de Supercomputacao, Universidade Federal do Rio Grande do Sul (CESUP/UFRGS)

NCP-1

Hypercube NCP-1 (8 nodes each with a T800 and Intel i860).

Locale: Universidade Federal do Rio de Janeiro (COPPE/UFRJ)

Wavetracer

Wavetracer / SIMD

Locale: Universidade Federal de Minas Gerais (DCC/UFMG)

Appendix B

We list below some of the WWW sites related to Latin American universities and R&D related institutions.

Table 8: World Wide Web sites

Argentina (top level domain)	http://www.ar:70/
Universidad de Chile	http://www.dcc.uchile.cl/
Costa Ricas Research Network	http://ns.cr/
Ecuador: Universidade San Francisco de Quito	http://mail.usfq.edu.ec/
Mexico (Info)	http://info.pue.udlap.mx/
Pont. Univ. Catolica Rio de Janeiro (PUC/RJ)	http://www.puc-rio.br/
Univ. Federal do Rio de Janeiro (COPPE/UFRJ)	http://guarani.cos.ufrj.br:8000/
Univ. Federal de Minas Gerais	http://dcc.ufmg.br/
Univ. Federal do Rio Grande do Sul (Inst. Inf.)	http://tucano.inf.ufrgs.br/
Univ. Federal do Rio Grande do Sul (CESUP)	http://www.cesup.ufrgs.br/
Univ. Federal de Santa Catarina	http://www.inf.ufsc.br/
Universidade Estadual de Campinas	http://www.unicamp.br/
Universidade de Sao Paulo	http://www.usp.br/
	http://www.lsi.usp.br/
FAPESP Fund. de Amparo a Pesq. Est. S. Paulo	http://www.fapesp.br/
CNPq Conselho Nacional de Des. Cient. e Tecnol.	http://www.cnpq.br/
CITE/CNPQ (Softex 2000, ProTem, RNP)	http://www-cite.cnpq.br/
INPE - Inst. Nac. de Pesquisas Espaciais	http://www.inpe.br/
LNCC - Lab. Nac. Computacao Cient.	http://www.lncc.br/

References

[1] Agarwal, A. et al. *"The mit alewife machine: architecture and performance"*. **Proceedings of the 22nd International Symposium on Computer Architecture**. June 1995.

[2] Amorim, C. L. et al. *"O sistema de computação paralela NCP I"*. **Anais do 7.o Simposio Brasileiro de Arquitetura de Computadores e Processamento Paralelo**. July 1993, pp. 89–99.

[3] Baeza-Yates, R. A., Fuller, D. A., Pino, J. A. and Goodman, S. E. *"Computing in Chile: the Jaguar of the Pacific Rim?"*. **Communications of the ACM**, Vol. 38, No. 9, pp. 23–28, 1995.

[4] Bershad, B. N., Zekauskas, M. J. and Sawdon, W. A. *"The midway distributed shared memory system"*. **Proceedings of COMPCON**, Spring 1993.

[5] Cordsen, J. *"Basing virtually shared memory on a family of consistency models"*. **Proceedings of the International Workshop on Support for Large Scale Shared Memory Architectures**, pp. 58–72, April 1994.

[6] Cordsen, J. and Gubitoso, M. D. *"Performance considerations in a virtually shared memory system"*. Technical Report – PLENA project. GMD – FIRST and Universidade de São Paulo. September, 1995.

[7] Dedrick, J. L. *"Little Engines that Could: Computing in Small Energetic Countries"*. **Communications of the ACM**, Vol. 38, No. 5, pp. 21–26, 1995.

[8] Dehne, F., Fabri, A. and Rau-Chaplin, A. *"Scalable parallel computational geometry for coarse grained multicomputers"*.**Proceedings of the ACM Symposium on Computational Geometry**, pp. 298–307, 1993.

[9] Dehne, F., Kenyon, C. and Fabri, A. *"Scalable and Architecture Independent Parallel Geometric Algorithms with High Probability Optimal Time"*, **Proceedings of the IEEE Symposium on Parallel and Distributed Processing**, Dallas, Tx, pp. 586–593, 1994.

[10] Dehne, F. and Song, S. W. *"Scalable Parallel Pointer Jumping for Distributed Memory Multiprocessors, and Applications"*. Technical Report, Universidade de São Paulo, Instituto de Matemática e Estatística, Departamento de Ciencia da Computação. In preparation.

[11] Dongarra, J. J., Meuer, H. W. and Strohmaier, E. *"TOP500 Supercomputer Sites"*. Technical Report RUM 33/93. Computing Centre, University of Mannheim. 11 November 1993.

[12] Dongarra, J. J., Meuer, H. W. and Strohmaier, E. *"TOP500 Supercomputer Sites"*. Technical Report RUM 42/95. Computing Centre, University of Mannheim. 10 November 1995.

[13] EITO European Information Technology Observatory, 1994.

[14] Estivill-Castro, V. *"Computer Science in Mexico"*. **IEEE Computer**, Vol. 28, No. 8, pp. 56–62, 1995.

[15] Gubitoso, M. D. and Cordsen, J. *"Performance considerations in VOTE for PEACE"*. Technical Report RT-MAC-9514, Universidade de São Paulo, Instituto de Matemática e Estatística, Departamento de Ciência da Computação. November 1995.

[16] Kuskin, J. et al. *"The flash multiprocessor"*. **Proceedings of the 21st International Symposium on Computer Architectures**. April 1994.

[17] La Rovere, R. L. and Goodman, S. E. *"Computing in the Brazilian Amazon"*. **Communications of the ACM**, Vol. 35, No. 4, pp. 21–24, 1992.

[18] Lux, T., Sydow, A. and Schmidt, M. *"An Air Pollution Model Used for the Region of Berlin"*. **Proceedings of the 7th International Conference on Computer Methods for Partial Differential Equations**, June 1992.

[19] Rosenthal, D. and Meira, S. L. M. *Os Primeiros 15 Anos da Politica Nacional de Informática: o Paradigma e sua Implementação*. CNPq/ProTem-CC – Conselho Nacional de Desenvolvimento Científico e Tecnológico/Programa Tematico em Cincia da Computação. 285 p. 1995.

[20] Softex 2000, CITE/CNPq (Coordination of Technology Innovation, Conselho Nacional de Desenvolvimento Cientifico e Tecnologico) – http://www-cite.cnpq.br/

V. CONCLUSIONS

by

Boleslaw K. Szymanski

Several conclusions can be drawn from the current state of parallel processing. Computer hardware and software are products with very unusual cost structures. The bulk of the funds is spent at the design and implementation stages, including tooling costs for hardware. The cost of production (replication would be a better word to use in this case), is relatively small for hardware and virtually insignificant for software. This cost structure implies that the price of the individual product is almost inversely proportional to the number of products produced. As a result, products for a vast market are orders of magnitude less expensive than comparable products for narrow markets. An example of this can be seen in the case of software systems. A compiler of a modern laguage, such as C++, costs in the order of US$ 100 for personal computers, but a comparable ADA compiler for a workstation is likely to cost several thousand dollars.

Currently, the parallel processing market is only a tiny fraction of the overall computer market. Consequently, anything targeted specifically to the small parallel market is several times more expensive than the comparable product mass produced for the general computer market – and the economies of scale have not gone unnoticed by the hardware producers! In the last few years, we have witnessed a shift from custom design chips used in earlier parallel machines, such as KSR, Connection Machine, to COTS (commercial off-the-shelf) chips in modern parallel computers (examples are the SP series from IBM and the Challenger series from Silicon Graphics). These newer machines not only enjoy better price-performance ratios than their predecessors, but they are also readily upgradable following improvements to the mainstream hardware. A good example of such a transition was the change from the SP-1 to SP-2 series in 1994, which resulted in doubling the speed of a single processor on the SP machines. Another advantage of this approach is the compatibility of the basic software for parallel and sequential machines using the same single processor architecture.

It is the opinion of the editor of this publication that a similar process will take place with software. More and more software for parallel machines will just be modified verisons of the mainstream software for sequential machines. This process will be hastened by the wider use of software engineering techniques that support the ease of program modifiability, such as object oriented technologies.

At the same time, parallel processing is becoming a mainstream tool in conducting and dispersing science, technology and information. The rapid process of using parallel processing in these areas cannot be ignored by the authorities responsible for science and technology without seriously jeapordising the future competitiveness of their country's industry and science base. These factors and many more described in the preceding chapters have led us to the following recommendations for governmental support in the area of parallel processsing:

1. Parallel processing must be funded as part of an overall governmental support for science and technology. With the exception of countries with the largest economies, such as Japan or the USA, the parallel processing market on a scale of a single country is too small to be self-supporting exclusively by industry. The particular percentage of the total science and technology funds that should be directed to support parallel processing is dependent on the particular mix of industries of the country in question, the importance of national defense and the level of technological development of each country. However, the general range would usually be in the order of a few per cent of the total spending for science and technology.

2. The resources for parallel processing should be directed mainly towards building a network of computing centres and educating their users. The preceeding chapters of this publication describe the encouraging examples of such networks both in Asia and South America. Access to parallel computers and the education of the users are the most important ingredients of a successful policy. Such a policy can pay back by creating a community of users and a set of applications that will form the kernel of future expertise in parallel processing. In such an environment, the applications are likely to grow in value and importance, leading to self-support of parallel processing and a decrease of its dependence on governmental funds.

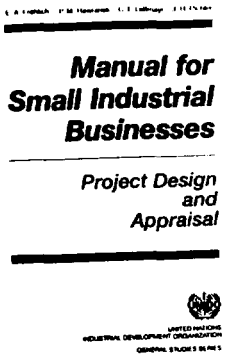
3. Building the set of applications encourages the development of parallel programming tools of general use. Hence, support for parallel software research is an important component of an overall national parallel processing funding. Such research must be application driven, because ultimately, applications and current technology are the measures of success or failure of such research. It is therefore important that such research will be integrated into the national network of parallel processing centres. Consequently, only a small fraction (10 to 20 per cent) of the parallel processing funding should be directed towards parallel software research. Again, examples from Brazil and India presented in the earlier chapters indicate that such a course of action is already underway.

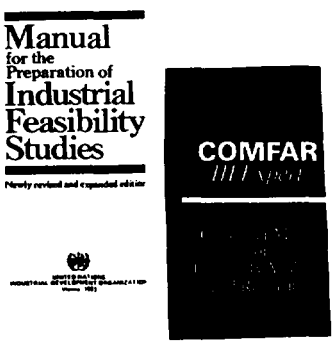
4. The economy of scale also underscores the importance of international cooperation in the area of parallel processing. Hence, it is desirable to devote limited funds for such cooperation to enable local users to receive the necessary education or to participate in conferences abroad, while at the same time enabling the government to secure the services of foreign experts.

5. The final recommendation concerns parallel hardware design. On the basis of scale, it is not economically viable to support the development of parallel hardware from national parallel processing funds. Of course, other reasons, such as access to the most modern hardware and national security can override economic concern. However, excepting such special circumstances, the parallel processing funding is far better used if it is spent on goals outlined in points 2 and 3 above. No country is identical to any other, so a parallel processing support policy for each individual country must be carefully designed, based on the broad guidelines as described above, and on an expert evaluation of the particulars of the social, economical and political situation of the country in question.

INVESTMENT
AND TECHNOLOGY
PARTNERSHIP
INITIATIVE

INVESTMENT AND TECHNOLOGY PARTNERSHIP INITIATIVE

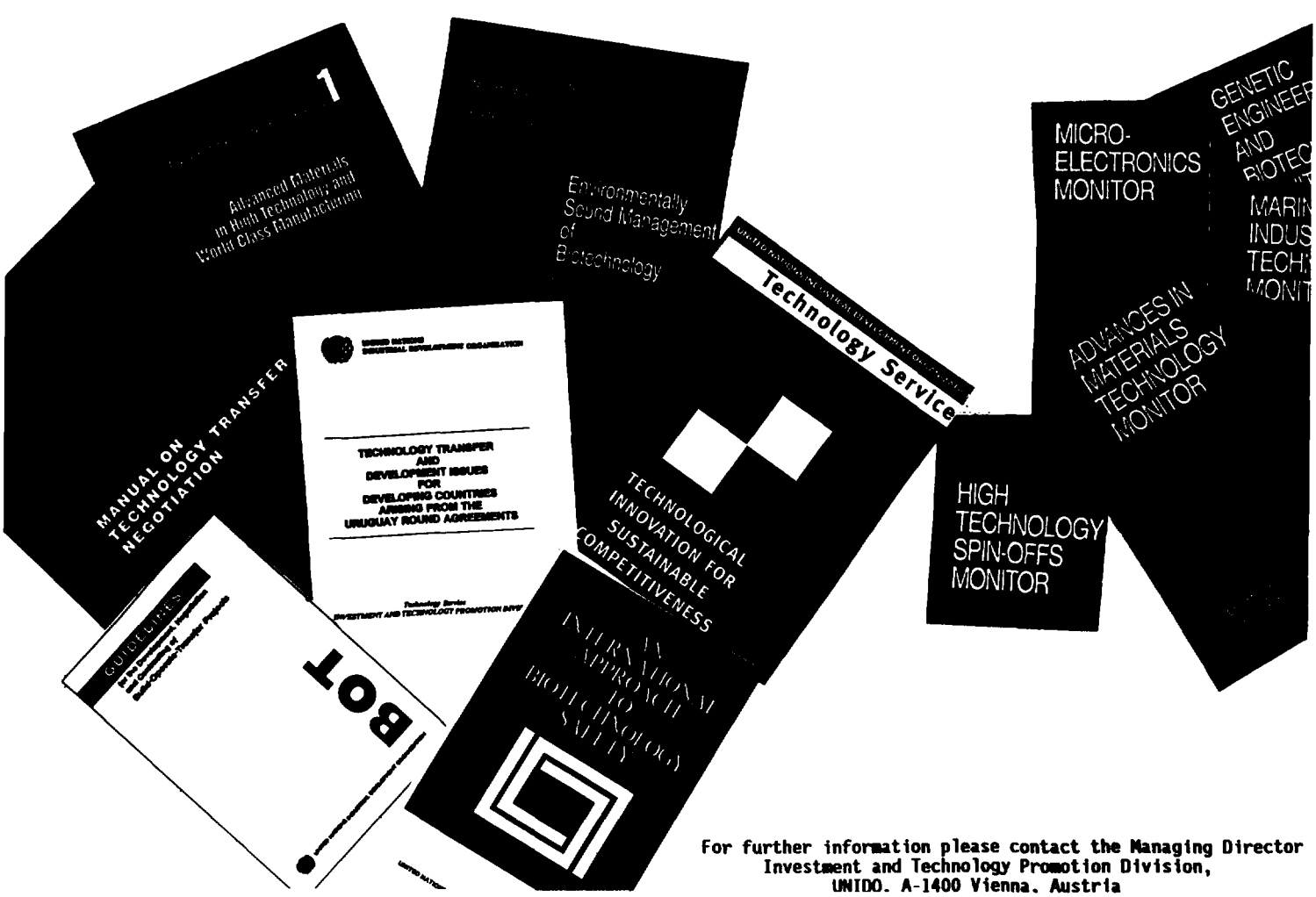




Investment and technology play a vital role in the industrial growth of developing countries as well as in their gradual integration into the international economy. However, the changing national and international economic environments are prompting developing countries to take a fresh look at their efforts to secure and widen investment and technology flows.

While UNIDO is one of several institutions promoting investment to developing countries, it is the only one with a comprehensive background in industrial development and technology transfer. UNIDO can therefore be instrumental in linking investment promotion with technology upgrading and the enhancement of technological capabilities. UNIDO is in the position to provide a wide range of services to facilitate developing countries in gaining access to investment resources, technology and know-how, marketing and managerial skills as well as in upgrading their domestic capabilities in these fields.

SELECTED UNIDO PUBLICATIONS RELATED TO TECHNOLOGY PROGRAMME



For further information please contact the Managing Director
Investment and Technology Promotion Division,
UNIDO, A-1400 Vienna, Austria