High Performance Distributed Computing

Geoffrey C. Fox
gcf@npac.syr.edu
http://www.npac.syr.edu

Northeast Parallel Architectures Center
111 College Place
Syracuse University
Syracuse, New York 13244-4100

Abstract

High Performance Distributed Computing (HPDC) is driven by the rapid advance of two related technologies—those underlying computing and communications, respectively. These technology pushes are linked to application pulls, which vary from the use of a cluster of some 20 workstations simulating fluid flow around an aircraft, to the complex linkage of several hundred million advanced PCs around the globe to deliver and receive multimedia information. The review of base technologies and exemplar applications is followed by a brief discussion of software models for HPDC, which are illustrated by two extremes—PVM and the conjectured future World Wide Web based WebWork concept. The narrative is supplemented by a glossary describing the diverse concepts used in HPDC.

1 Motivation and Overview

Advances in computing are driven by VLSI or very large scale integration, which technology has created the personal computer, workstation, and parallel computing markets over the last decade. In 1980, the Intel 8086 used 50,000 transistors while today's (1995) "hot" chips have some five million transistors——a factor of 100 increase. The dramatic improvement in chip density comes together with an increase in clock speed and improved design so that today's workstations and PCs (depending on the function) have a factor of 50—1,000 better performance than the early 8086 based PCs. This performance increase enables new applications. In particular, it allows real time multimedia decoding and display—this capability will be exploited in the next generation of video game controllers and set top boxes, and be key to implementing digital video delivery to the home.

The increasing density of transistors on a chip follows directly from a decreasing feature size which is in 1995 $0.5\,\mu$ for the latest Intel Pentium. Feature size will continue to decrease and by the year 2000, chips with 50,000,000 transistors are expected to be available.

Communications advances have been driven by a set of physical transport technologies that can carry much larger volumes of data to a much wider range of places. Central is the use of optical fiber, which is now competitive in price with the staple twisted pair and coaxial cable used by the telephone and cable industries, respectively. The widespread deployment of optical fibers also builds on laser technology to generate the light, as well as the same VLSI advances that drive computing. The latter are critical to the high-performance digital switches needed to route signals between arbitrary source, and destination. Optics is not the only physical medium of importance—continuing and critical communications advances can be assumed for satellites and wireless used for linking to mobile (cellular) phones or more generally the future personal digital assistant (PDA).

One way of exploiting these technologies is seen in parallel processing. VLSI is reducing the size of computers, and this directly increases the performance because reduced size corresponds to increased clock speed, which is approximately proportional to $(1/\lambda)$ for feature size λ. Crudely, the cost of a given number of transistors is proportional to silicon used or λ^2, and so the cost performance improves by $(1/\lambda)^3$. This allows personal computers and workstations to deliver today, for a few thousand dollars, the same performance that required a supercomputer costing several million dollars just ten years ago. However, we can exploit the technology advances in a different way by increasing performance instead of (just) decreasing cost. Here, as illustrated in Figure 1, we build computers consisting of several of the basic VLSI building blocks. Integrated parallel computers require high speed links between the individual processors, called nodes. In Figure 1, these links are etched on a printed circuit board, but in larger systems one would also use cable or perhaps optical fiber to interconnect nodes. As shown in Figure 2, parallel computers have become the dominant supercomputer technology with current high and systems capable of performance of up to 100 GigaFLOPS or 10^{11} floating point operations per second. One expects to routinely install parallel machines capable of TeraFLOPS sustained performance by the year 2000.

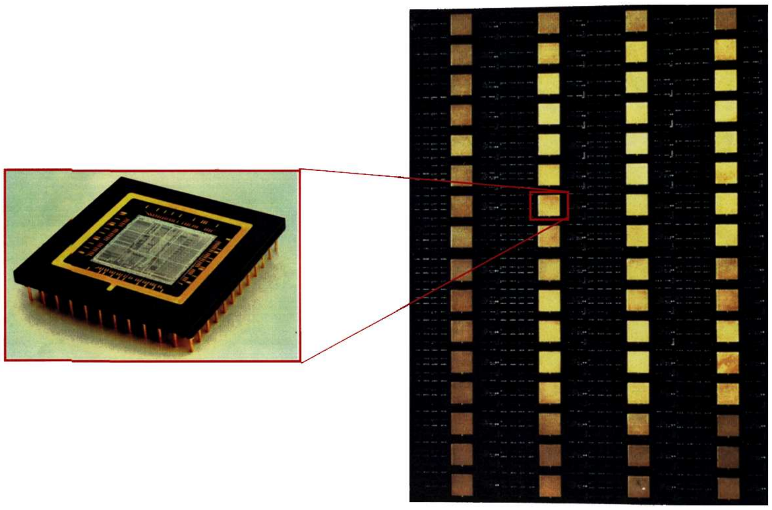

Figure 1: The nCUBE-2 Node and Its Integration into a Board. Up to 128 of these boards can be combined into a single supercomputer.

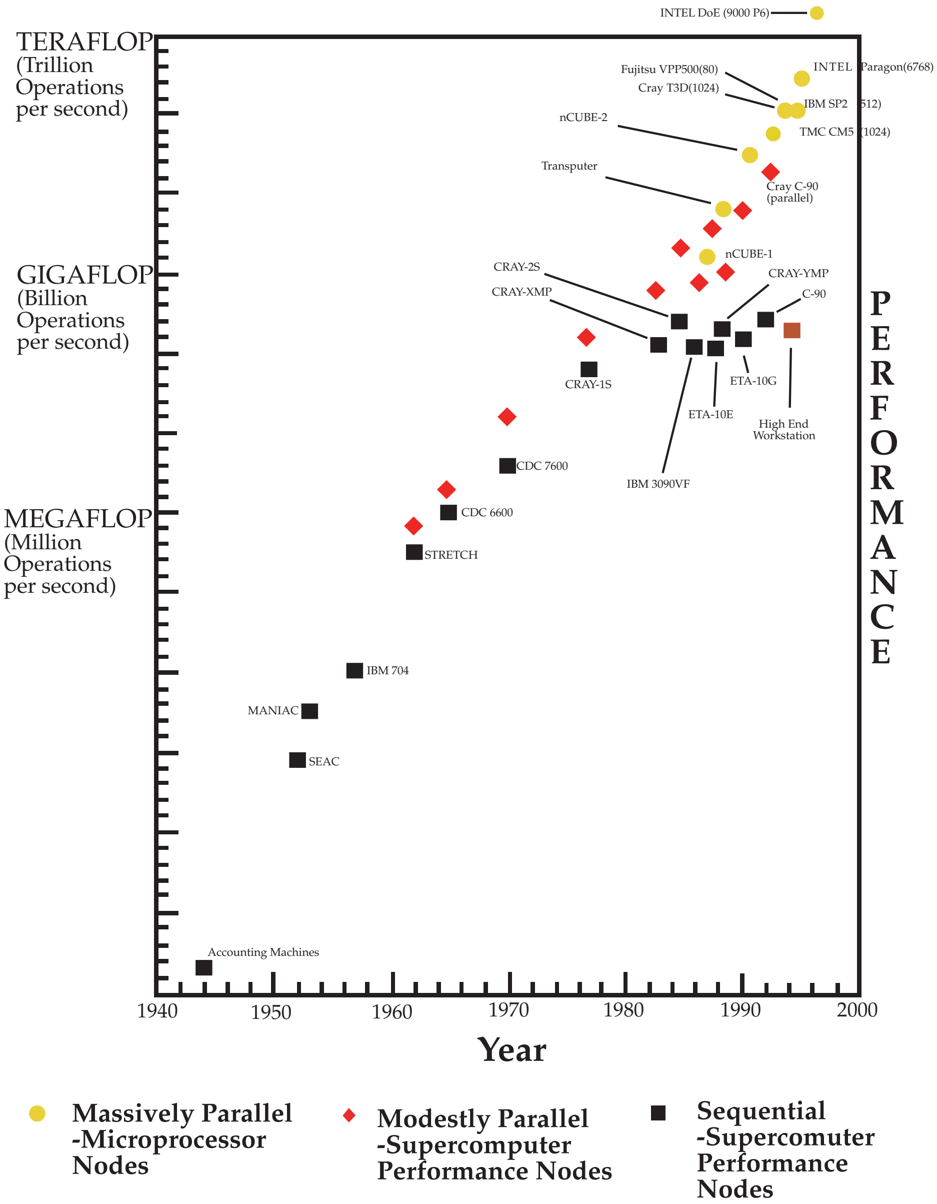

Figure 2: Performance of Parallel and Sequential Supercomputers

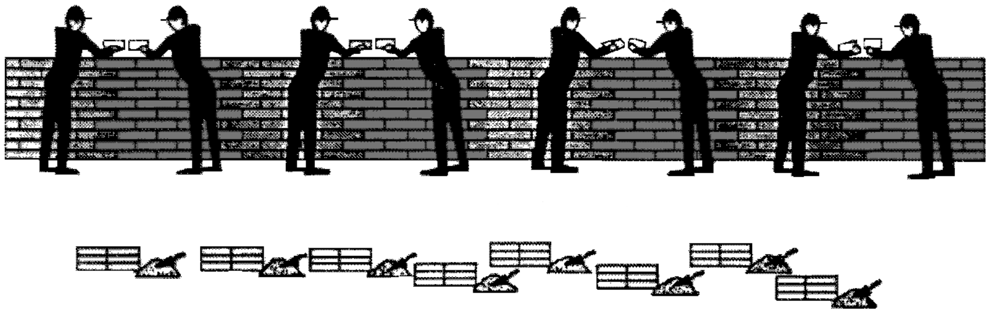

Figure 3: Concurrent Construction of a Wall using $N = 8$ Bricklayers

Often, one compares such highly coupled parallel machines with the human brain, which achieves its remarkable capabilities by the linkage of some 10^{12} nodes—neurons in the case of the brain—to solve individual complex problems, such as reasoning and pattern recognition. These nodes have individually mediocre capabilities and slow cycle time (about .001 seconds), but together they exhibit remarkable capabilities. Parallel (silicon) computers use fewer faster processors (current MPPs have at most a few thousand microprocessor nodes), but the principle is the same.

However, society exhibits another form of collective computing whereby it joins several brains together with perhaps 100,000 people linked in the design and production of a major new aircraft system. This collective computer is "loosely-coupled"—the individual components (people) are often separated by large distances and with modest performance "links" (voice, memos, etc.).

Figure 3 shows HPDC at work in society with a bunch of mason's building a wall. This example is explained in detail in [Fox:88a], while Figure 4 shows the parallel neural computer that makes up each node of the HPDC system of Figure 3.

We have the same two choices in the computer world. Parallel processing is gotten with a tightly coupled set of nodes as in Figure 1 or Figure 4. High Performance Distributed Computing, analogously to Figure 3, is gotten from a geographically distributed set of computers linked together with "longer wires," but still coordinated (a software issue discussed later) to solve a "single problem" (see application discussion later). HPDC is also known as metacomputing, NOW's (Network of Workstations) and COW's (Clusters of

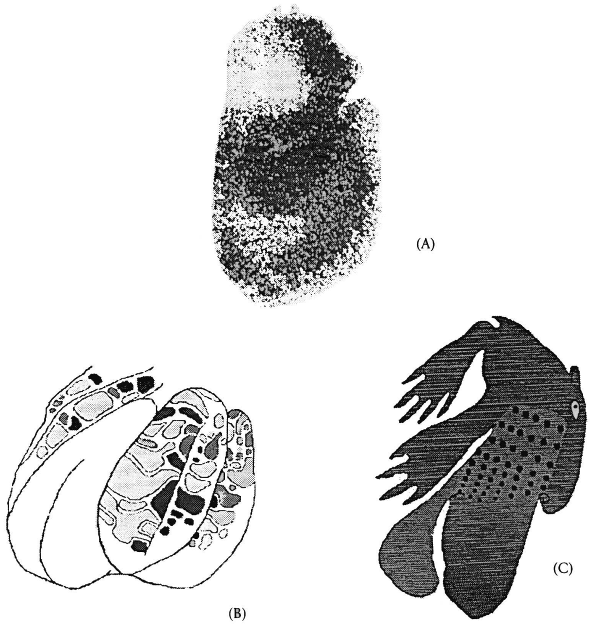

Figure 4: Three Parallel Computing Strategies Found in the Brain (of a Rat). Each figure depicts brain activity corresponding to various functions: (A) continuous map of a tactile inputs in somatosensory cortex, (B) patchy map of tactile inputs to cerebellar cortex, and (C) scattered mapping of olfactory cortex as represented by the unstructured pattern of 2DG update in a single section of this cortex [Nelson:90b].

Workstations), where each acronym has a slightly different focus in the broad HPDC area. Notice the network (the so called "longer wires" above) which links the individual nodes of an HPDC metacomputer can be of various forms—a few of the many choices are a local area network (LAN), such as ethernet or FDDI, a high-performance (supercomputer) interconnect, such as HIPPI or a wide area network WAN with ATM technology. The physical connectivity can be copper, optical fiber, and/or satellite. The geographic distribution can be a single room; the set of computers in the four NSF supercomputer centers linked by the vBNS; most grandiosely, the several hundred million computers linked by the Global Information Infrastructure (GII) in the year 2010.

HPDC is a very broad field—one can view client-server enterprise computing and parallel processing as special cases of it. As we describe later, it exploits and generalizes the software built for these other systems.

By definition, HPDC has no precise architecture or implementation at the hardware level—one can use whatever set of networks and computers is available to the problem at hand. Thus, in the remainder of the article, we focus first on applications and then some of the software models.

In the application arena, we go through three areas—aircraft design and manufacture, military command and control, and multimedia information systems. In each case, we contrast the role of parallel computing and HPDC—parallel computing is used for the application components that can be broken into modules (such as grid points for differential equation solvers or pixels for images), but these are linked closely in the algorithm used to manipulate them. Correspondingly, one needs the low latency and high internode communication bandwidth of parallel machines. HPDC is typically used for coarser grain decompositions (e.g., the different convolutions on a single image rather than the different blocks of pixels in an image). Correspondingly larger latencies, and lower bandwidths can be tolerated. HPDC and parallel computing often deal with similar issues of synchronization and parallel data decomposition, but with different tradeoffs and problem characteristics. Indeed, there is no sharp division between these two concepts, and clusters of workstations can be used for large scale parallel computing (as in CFD for engine simulation at Pratt and Whitney) while tightly coupled MPPs can be used to support multiple uncoupled users—a classic "embarrassingly parallel" HPDC application.

2 Applications

We have chosen three examples to discuss HPDC and contrast distributed and parallel computing. In the first and second, manufacturing and command and control, we see classic parallel computing for simulations, and signal processing respectively linked to geographically distributed HPDC. In the last, InfoVISiON, the parallel computing is seen in parallel multimedia databases, and the distributed HPDC aspects are more pronounced.

2.1 Manufacturing and Computational Fluid Dynamics

HPDC is used today, and can be expected to play a growing role in manufacturing, and more generally, engineering. For instance, the popular concept of agile manufacturing supposes the model where virtual corporations generate "products-on-demand." The NII is used to link collaborating organizations. HPDC is needed to support instant design (or more accurately redesign or customization) and sophisticated visualization and virtual reality "test drives" for the customer. At the corporate infrastructure level, concurrent engineering involves integration of the different component disciplines—such as design, manufacturing, and product life cycle support—involved in engineering. These general ideas are tested severely when they are applied to the design and manufacturing of complex systems such as automobiles, aircraft, and space vehicles such as shuttles. Both the complexity of these products, and in some sense the maturity of their design, places special constraints and challenges on HPDC.

High-performance computing is important in all aspects of the design of a new aircraft. However, it is worth noting that less than 5% of the initial costs of the Boeing 777 aircraft were incurred in computational fluid dynamics (CFD) airflow simulations—the "classic" Grand Challenge in this field. On the other hand, over 50% of these sunk costs could be attributed to overall systems issues. Thus, it is useful but not sufficient to study parallel computing for large scale CFD. This is "Amdahl's law for practical HPDC." If only 5% of a problem is parallelized, one can at best speed up and impact one's goals—affordability, time to market—by this small amount. HPDC, thus, must be fully integrated into the entire engineering enterprise to be effective. Very roughly, we can view the ratios of 5% to 50% as a measure of ratio of 1:10 of the relevance of parallel and distributed computing in this case.

The maturity of the field is illustrated by the design criterion used today. In the past, much effort has been spent on improving performance—more speed, range, altitude, size. These are still critical under extreme conditions, but basically these just form a given design framework that suffices to buy you a place at the table (on the short-list). Rather, the key design criteria is competitiveness, including time to market, and total affordability. Although the design phase is not itself a major cost item, decisions made at this stage lock in most of the full life cycle cost of an aircraft with perhaps 80% of total cost split roughly equally between maintenance and manufacturing. Thus, it certainly would be important to apply HPDC at the design phase to both shorten the design cycle (time to market) and lower the later ongoing costs of manufacturing and maintenance.

We take as an example the design of a future military aircraft—perhaps 10 years from now. This analysis is taken from a set of NASA sponsored activities centered on a study of ASOP—Affordable Systems Optimization Process. This involved an industrial team, including Rockwell International, Northrop Grumman, McDonnell Douglas, General Electric, and General Motors. ASOP is one of several possible approaches to multidisciplinary analysis and design (MAD) and the results of the study should be generally valid to these other MAD systems. The hypothetical aircraft design and construction project could involve six major companies and 20,000 smaller subcontractors. This impressive virtual corporation would be very geographically dispersed on both a national and probably international scale. This project could involve some 50 engineers at the first conceptual design phase. The later preliminary and detailed design stages could involve 200 and 2,000 engineers, respectively. The design would be fully electronic and demand major computing, information systems, and networking resources. For instance, some 10,000 separate programs would be involved in the design. These would range from a parallel CFD airflow simulation around the plane to an expert system to plan location of an inspection port to optimize maintainability. There is a corresponding wide range of computing platforms from PCs to MPPs and a range of languages from spreadsheets to high-performance Fortran. The integrated multidisciplinary optimization does not involve blindly linking all these programs together, but rather a large number of suboptimizations involving at one time a small cluster of these base programs. Here we see clearly, an essential role of HPDC to implement these set of geographically distributed optimizations. However, these clusters could well need linking of geographically separated compute and information systems. An aircraft is, of course, a very precise system, which

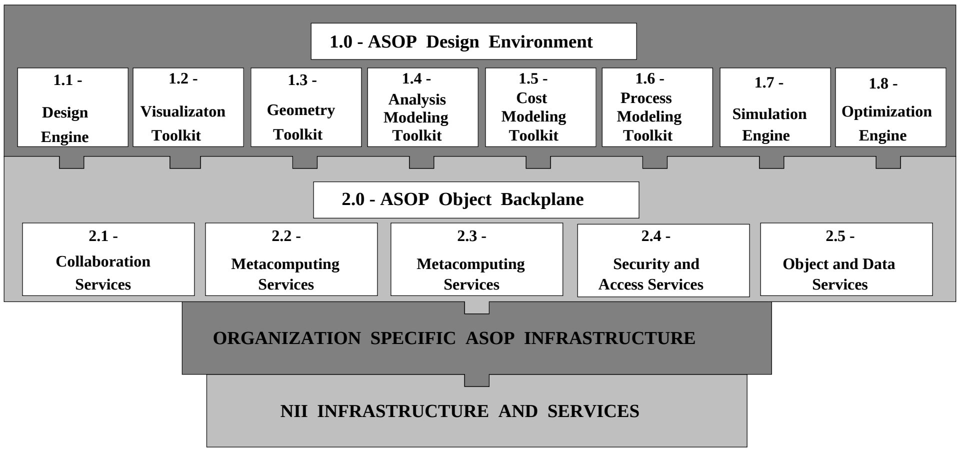

Figure 5: Affordable Systems Optimization Process (ASOP) Implemented on the NII for Aeronautics Systems

must work essentially flawlessly. This requirement implies a very strict coordination and control of the many different components of the aircraft design. Typically, there will be a master systems database to which all activities are synchronized at regular intervals–perhaps every month. The clustered sub-optimizations represent a set of limited excursions from this base design that are managed in a loosely synchronous fashion on a monthly basis. The configuration management and database system are both critical and represent a major difference between manufacturing and command and control, where in the latter case, real time "as good as you can do" response, is more important than a set of precisely controlled activities. These issues are characteristic of HPDC where, although loosely coupled, the computers on our global network are linked to "solve a single problem."

ASOP is designed as a software backplane (the NII) linking eight major services or modules shown in Figure 5. These are design (process controller) engine, visualization, optimization engine, simulation engine, process (manufacturing, productibility, supportability) modeling toolkit, costing toolkit, analytic modeling toolkit, and geometry toolkit. These are linked to a set of databases defining both the product and also the component properties. Parallel computing is important in many of the base services, but HPDC is seen in the full system.

2.2 Command and Control

Command Control (sometimes adding in Computing, Communications, Intelligence Surveillance, and Battle Management with abbreviations lumped together as BMC^4IS) is the task of managing and planning a military operation. It is very similar to the civilian area of Crisis management, where the operations involve combating effects of hurricanes, earthquakes, chemical spills, forest fires, etc. Both the military and civilian cases have computational "nuggets" where parallel computing is relevant. These include processing sensor data (signal and image processing) and simulations of such things as expected weather patterns and chemical plumes. One also needs large-scale multimedia databases with HPDC issues related to those described for InfoVISiON in Section 2.3.

HPDC is needed to link military planners and decision makers, crisis managers, experts at so-called anchor desks, workers (warriors) in the field, information sources such as cable news feeds, and large-scale database and simulation engines.

A key characteristic of the required HPDC support is adaptivity. Crises and battles can occur anywhere and destroy an arbitrary fraction of the existing infrastructure. Adaptivity means making the best use of the remaining links, but also deploying and integrating well mobile enhancements. The information infrastructure must exhibit security and reliability or at least excellent fault tolerance (adaptivity). Network management must deal with the unexpected capacity demands and real time constraints. Priority schemes must allow when needed critical information (such as the chemical plume monitoring and military sensor data) precedence over less time critical information, such as background network video footage.

Needed computing resources will vary from portable handheld systems to large backend MPPs. As there will be unpredictable battery (power) and bandwidth constraints, it is important that uniform user interfaces and similar services be available on all platforms with, of course, the fidelity and quality of a service reflecting the intrinsic power of a given computer. As with the communications infrastructure, we must cope with unexpected capacity demands. As long as the NII is deployed nationally, computational capacity can be exploited in remote sites. The Department of Defense envisages using the basic NII (GII) infrastructure for command and control, augmented by "theater extensions" to bring needed communications into critical areas. The "take it as it is" characteristic of command and control requires that operating systems and programming models support a general adaptive mix (metacomputer) of coordinated geographically distributed but networked computers. This network will adaptively link available people (using perhaps personal digital assistants) to large-scale computation on MPPs and other platforms. There are large computational requirements when forecasting in real-time physical phenomena, such as the weather effects on a projected military action, forest fires, hurricanes, and the structure of damaged buildings. On a longer time scale, simulation can be used for contingency planning and capability assessment. Training with simulated virtual worlds supporting televirtuality, requires major computational resources. In the information arena, applications include datamining to detect anomalous entries (outliers) in large federated multimedia databases. Data fusion including sensor input and processing, geographical information systems (with perhaps three-dimensional terrain rendering), and stereo reconstruction from multiple video streams are examples of compute intensive image processing forming part of the needed HPDC environment.

A critical need for information management involves the best possible high-level extraction of knowledge from databanks—the crisis manager must make judgments in unexpected urgent situations—we cannot carefully tailor and massage data ahead of time. Rather, we need to search a disparate set of multimedia databases. As well as knowledge extraction from particular information sources, the systematic use of metadata allowing fast coarse grain searching is very important. This is a specific example of the importance of standards in expediting access to "unexpected" databases. One requires access to databases specific to crisis region or battlefield, and widespread availability of such geographic and community information in electronic form is essential. There are very difficult policy and security issues, for many of these databases need to be made instantly available in a hassle-free fashion to the military commander or crisis manager—this could run counter to proprietary and security classification constraints. The information system should allow both network news and warriors in the field to deposit in near real-time, digital versions of their crisis and battlefield videos and images.

As mentioned, we expect that human and computer expertise to be available in "anchor desks" to support instant decisions in the heat of the battle. These have been used in a set of military exercises called JWID (Joint Warrior Interoperability Demonstrations). We note that this information scenario is a real-time version of that described in the next section as InfoVISiON to support the society of the Information Age.

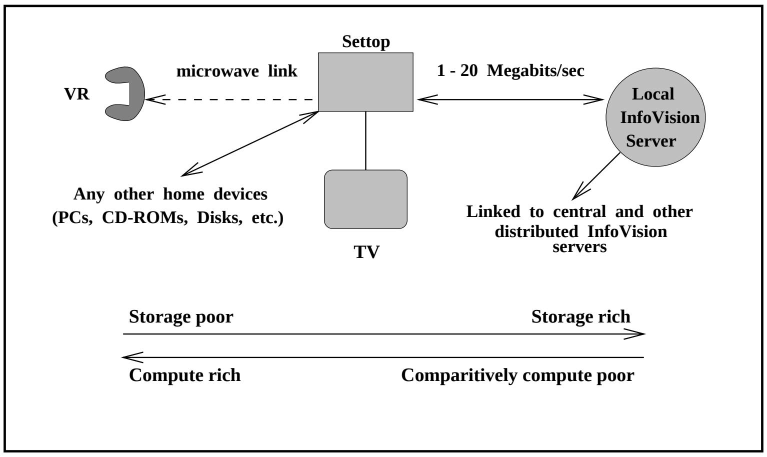

Figure 6: The basic *InfoVISiON* scenario as seen by a home in the year 2000 with an intelligent settop box interfacing the digital home to a hierarchical network of InfoVISiON servers

Command and Control has historically used HPDC as the relevant computer and communication resources, are naturally distributed, and not centralized into a single MPP. We see this HPDC model growing into the standard information support environment for all the nation's enterprises, including business, education, and society. We now explore this in the following section.

2.3 Application Exemplar—InfoVISiON

High-performance distributed computers solve problems in science and engineering. We think of these problems as simulations of airflow, galaxies, bridges, and such things. However, presumably entertaining, informing, and educating society is an equally valid problem. Computers and networks of the NII will be able (see Figure 6) to deliver information at many megabits/second to "every" home, business (office), and school "desk." This network can be considered as an HPDC system because one expects the information to be stored in a set of distributed multimedia services that could vary from PCs to large MPPs and be delivered to a larger set of clients. As shown in Figure 7, one can estimate that the total compute capability in these servers and clients will be some hundred times greater than that of the entire set of supercomputers in the nation.

NII Compute & Communications Capability in Year 2005 - 2020

100 Supercomputers at a Teraflop each	10^{14} (Fl)ops/sec at 100% Duty Cycle
100 Million NII Offramps or Connections at realtime video speeds	10^{14} bits to words/sec at about 10% to 30% Duty Cycle
100 Million home PCs, Videogames or Settop Boxes at 100-1000 Mega(Fl)ops each	10^{16} to 10^{17} (Fl)ops/sec at about 10% to 30% Duty Cycle
1,000 to 10,000 High Performance Multimedia (parallel) servers each with some 1,000 to 10,000 nodes	10^{15} to 10^{16} ops/sec at 100% Duty Cycle

Each of three components (network connections, clients, servers) has capital value of order $10 to $100 billion

Figure 7: An estimate of the communication bandwidth and compute capability contained in the NII and supercomputer industries

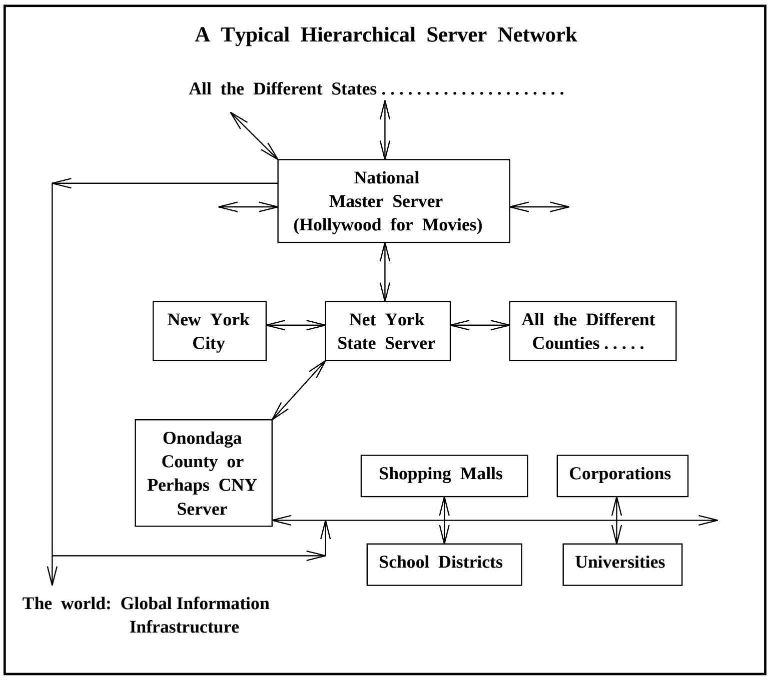

Figure 8: A typical hierarchical server network depicted for a master system in Hollywood cascading down with a fragment of node systems shown for central New York

The computational issues in this application are somewhat different than those for the previous cases we considered. Classic data parallelism and languages, such as High Performance Fortran, are not critical. Large-scale distributed databases are the heart of this application, which are accessed through the exploding set of Web technologies. Presumably, these will need to migrate from today's clients (PC/workstations) to more user friendly, and at least initially less flexible settop box implementations controlling home entertainment systems. We will find the same need for data locality as in large scale simulations. As shown in Figure 8, when the latest Hollywood movie is released on the NII, one will not have half the nation directly connected to Hollywood. Rather, data is automatically replicated or cached on local servers so that one will only need to communicate such "hot" information over a distance of a few miles. As in simulation examples, communication bandwidth will be limited and such steps are needed to reduce demand.

InfoVISiON will require simulation, but it will be more loosely coupled than for say large-scale CFD, and consist of very many smaller problems. Interactive videogaming with multiple players sharing a virtual world is one clear need for simulation on the NII, and for this the three-dimensional database VRML has been introduced. However, another example that can use the same technology is remote viewing and exploration of consumer products, such as cars, furniture, and large appliances. Simulation will support virtual reality like exploration and the automatic customization of such products for particular customers.

3 Software for HPDC

As in all areas of computing, the software support for HPDC is built up in a layered fashion with no clearly agreed architecture for these layers. We will describe software for a target metacomputer consisting of a set of workstations. The issues are similar if you include more general nodes, such as PCs or MPPs. Many of the existing software systems for HPDC only support a subset of possible nodes, although "in principle" all could be extended to a general set.

In a layered approach, we start with a set of workstations running a particular variant of UNIX with a communication capability set up on top of TCP/IP. As high-speed networks such as ATM become available, there has been substantial research into optimizing communication protocols so as to take advantage of the increasing hardware capability. It seems likely that the flexibility of TCP/IP will ensure it will be a building block of all but the most highly optimized and specialized HPDC communication systems. On top of these base distributed computing operating system services, two classes of HPDC software support have been developed.

The most important of these we can call the MCPE (Metacomputing Programming Environment) which is the basic application program interface (API). The second class of software can be termed MCMS (Metacomputing Management System) which provides the overall scheduling and related services.

For MCMS software, the best known products are probably Load Leveler (from IBM produced for the SP-2 parallel machine), LSF (Load Sharing Facility from Platform Computing), DQS (Distributed Queuing System), and Condor from Wisconsin State University. Facilities provided by this software class could include batch and other queues, scheduling and node allocation, process migration, load balancing, and fault tolerance. None of the current systems is very mature and they do not offer integration of HPDC with parallel computing, and such standards as HPF or MPI.

We can illustrate two possible MCPE's or API's with PVM and the World Wide Web. PVM offers basic message passing support for heterogeneous nodes. It also has the necessary utilities to run "complete jobs" with appropriate processes spanned.

The World Wide Web is, of course, a much more sophisticated environment, which currently offers HPDC support in the information sector. However, recent developments, such as Java, support embedded downloaded applications. This naturally enables embarrassingly parallel HPDC computational problems. We have proposed a more ambitious approach, termed WebWork, where the most complex HPDC computations would be supported. This approach has the nice feature that one can integrate HPDC support for database and compute capabilities. All three examples discussed in Section 2 required this. One can be concerned that the World Wide Web will have too much overhead with its use of HTTP communication protocol, and substantial Web server processing overhead. We expect that the benefits of the flexibility of WebWork will outweigh the disadvantages of these additional overheads. We can also expect the major Web Technology development around the world to lead too much more efficient server and communication systems.

Glossary

Applets An application interface where referencing (perhaps by a mouse click) a remote application as a hyperlink to a *server* causes it to be downloaded and run on the *client*.

Asynchronous Transfer Mode (ATM) ATM is expected to be the primary networking technology for the *NII* to support multimedia communications. ATM has fixed length 53 byte messages (cells) and can run over any media with the cells asynchronously transmitted. Typically, ATM is associated with Synchronous Optical Network (SONET) optical fiber digital networks running at rates of OC-1 (51.84 megabits/sec), OC-3 (155.52 megabits/sec) to OC-48 (2,488.32 megabits/sec).

Bandwidth The communications capacity (measured in bits per second) of a transmission line or of a specific path through the network.

Clustered Computing A commonly found computing environment consists of many workstations connected together by a local area network. The workstations, which have become increasingly powerful over the years, can together, be viewed as a significant computing resource. This resource is commonly known as cluster of workstations, and can be generalized to a heterogeneous collection of machines with arbitrary architecture.

Command and Control This refers to the computer support decision making environment used by military commanders and intelligence officers. It is described in Section 2.2.

COW or NOW Clusters of Workstations (COW) are a particular HPDC environment where often one will use optimized network links and interfaces to achieve high performance. A COW—if homogeneous—is particularly close to a classic homogeneous MPP built with the same CPU chipsets as workstations. Proponents of COW's will claim that use of commodity workstation nodes allow them to track technology better than MPP's. MPP proponents note that their optimized designs deliver higher performance, which outweighs the increased cost of low-volume designs, and effective performance loss due to later (maybe only months) adoption of a given technology by the MPP compared to commodity markets.

Network of Workstations (NOW at http://now.cs.berkeley.edu/) and SHRIMP (Scalable High-Performance Really Inexpensive Multi Processor at http:/www.cs.princeton.edu/Shrimp/) are well-known research projects developing COWs.

Data Locality and Caching A key to sequential parallel and distributed computing is data locality. This concept involves minimizing "distance" between processor and data. In sequential computing, this implies "caching" data in fast memory and arranging computation to minimize access to data not in cache. In parallel and distributed computing, one uses migration and replication to minimize time a given node spends accessing data stored on another node.

Data Mining This describes the search and extraction of unexpected information from large databases. In a database of credit card transactions, conventional database search will generate monthly statements for each customer. Data mining will discover using ingenious algorithms, a linked set of records corresponding to fraudulent activity.

Data Parallelism A model of parallel or distributed computing in which a single operation can be applied to cell elements of a data structure simultaneously. Often, these structures are arrays.

Data Fusion A common *command and control* approach where the disparate sources of information available to a military or civilian commander or planner, are integrated (or fused) together. Often, a *GIS* is used as the underlying environment.

Distributed Computing The use of networked heterogeneous computers to solve a single problem. The nodes (individual computers) are typically loosely coupled.

Distributed Computing Environment The OSF Distributed Computing Environment (DCE) is a comprehensive, integrated set of services that supports the development, use and maintenance of distributed applications. It provides a uniform set of services, anywhere in the network, enabling applications to utilize the power of a heterogeneous network of computers. http://www.osf.org/dce/

Distributed Memory A computer architecture in which the memory of the nodes is distributed as separate units. Distributed memory hardware can support either a distributed memory programming model, such as message passing or a shared memory programming model.

Distributed Queuing System (DQS) An experimental UNIX based queuing system being developed at the Supercomputer Computations Research Institute (SCRI) at The Florida State University. *DQS* is designed as a management tool to aid in computational resource distribution across a network, and provides architecture transparency for both users and administrators across a heterogeneous environment. http://www.scri.fsu.edu/ pasko/dqs.html

Embarrassingly Parallel A class of problems that can be broken up into parts, which can be executed essentially independently on a parallel or distributed computer.

Geographical Information System (GIS) A user interface where information is displayed at locations on a digital map. Typically, this involves several possible overlays with different types of information. Functions, such as image processing and planning (such as shortest path) can be invoked.

Gigabit A measure of *network* performance—one Gigabit/sec is a bandwidth of 10^9 bits per second.

Gigaflop A measure of computer performance—one Gigaflop is 10^9 floating point operations per second.

Global Information Infrastructure (GII) The GII is the natural worldwide extension of the *NII* with comparable exciting vision and uncertain vague definition.

High-Performance Computing and Communications (HPCC) Refers generically to the federal initiatives, and associated projects and technologies that encompass *parallel computing*, *HPDC*, and the *NII*.

High-Performance Distributed Computing (HPDC) The use of distributed networked computers to achieve high performance on a single problem, i.e., the computers are coordinated and synchronized to achieve a common goal.

HPF A language specification published in 1993 by experts in compiler writing and parallel computation, the aim of which is to define a set of directives which will allow a Fortran 90 program to run efficiently on a distributed memory machine. At the time of writing, many hardware vendors have expressed interests, a few have preliminary compilers, and a few independent compiler producers also have early releases. If successful, HPF would mean data parallel programs can be written portably for various multiprocessor platforms.

Hyperlink The user level mechanism (remote address specified in a *HTML* or *VRML* object) by which remote services are accessed by *Web Clients* or *Servers.*

Hypertext Markup Language (HTML) A syntax for describing documents to be displayed on the *World Wide Web.*

Hypertext Transport Protocol (HTTP) The *protocol* used in the communication *Web Servers* and *clients.*

InfoVISiON Information, Video, Imagery, and Simulation ON demand is scenario described in Section 3 where *multimedia servers* deliver multimedia information to clients on demand—at the click of the user's mouse.

Integrated Service Data Network (ISDN) A digital multimedia service standard with a performance of typically 128 kilobits/sec, but with possibility of higher performance. ISDN can be implemented using existing telephone (*POTS*) wiring, but does not have the necessary performance of 1–20 megabits/second needed for full screen TV display at either VHS or high definition TV (HDTV) resolution. Digital video can be usefully sent with ISDN by using quarter screen resolution and/or lower (than 30 per second) frame rate.

Internet A complex set of interlinked national and global networks using the IP messaging protocol, and transferring data, electronic mail, and *World Wide Web.* In 1995, some 20 million people could access Internet—typically by *POTS.* The Internet has some high-speed links, but the majority of transmissions achieve (1995) bandwidths of at best 100 kilobytes/sec. the Internet could be used as the network to support a *metacomputer*, but the limited *bandwidth* indicates that *HPDC* could only be achieved for *embarrassingly parallel* problems.

Internet Protocol (IP) The network-layer communication protocol used in the DARPA Internet. IP is responsible for host-to-host addressing and routing, packet forwarding, and packet fragmentation and reassembly.

Java A distributed computing language (*Web Technology*) developed by Sun, which is based on C++ but supports *Applets*.

Latency The time taken to service a request or deliver a message which is independent of the size or nature of the operation. The latency of a *message passing* system is the minimum time to deliver a message, even one of zero length that does not have to leave the source processor. The latency of a file system is the time required to decode and execute a null operation.

LAN, MAN, WAN Local, Metropolitan, and Wide Area Networks can be made from any or many of the different physical network media, and run the different protocols. LAN's are typically confined to departments (less than a kilometer), MAN's to distances of order 10 kilometers, and WAN's can extend worldwide.

Loose and Tight Coupling Here, coupling refers to linking of computers in a network. Tight refers to low latency, high bandwidth; loose to high latency and/or low bandwidths. There is no clear dividing line between "loose" or "tight."

Massively Parallel Processing (MPP) The strict definition of MPP is a machine with many interconnected processors, where 'many' is dependent on the state of the art. Currently, the majority of high-end machines have fewer than 256 processors. A more practical definition of an MPP is a machine whose architecture is capable of having many processors—that is, it is scalable. In particular, machines with a distributed memory design (in comparison with shared memory designs) are usually synonymous with MPPs since they are not limited to a certain number of processors. In this sense, "many" is a number larger than the current largest number of processors in a shared-memory machine.

Megabit A measure of network performance—one Megabit/sec is a bandwidth of 10^6 bits per second. Note eight bits represent one character—called a byte.

Message Passing A style of inter-process communication in which processes send discrete messages to one another. Some computer architectures are called message passing architectures because they support this model in hardware, although message passing has often been used to construct operating systems and network software for sequential processors, shared memory, and distributed computers.

Message Passing Interface (MPI) The parallel programming community recently organized an effort to standardize the communication subroutine libraries used for programming on massively parallel computers such as Intel's Paragon, Cray's T3D, as well as networks of workstations. MPI not only unifies within a common framework programs written in a variety of exiting (and currently incompatible) parallel languages but allows for future portability of programs between machines.

Metacomputer This term describes a collection of heterogeneous computers networked by a high-speed wide area network. Such an environment would recognize the strengths of each machine in the Metacomputer, and use it accordingly to efficiently solve so-called *Metaproblems*. The *World Wide Web* has the potential to be a physical realization of a Metacomputer.

Metaproblem This term describes a class of problem which is outside the scope of a single computer architectures, but is instead best run on a Metacomputer with many disparate designs. These problems consist of many constituent subproblems. An example is the design and manufacture of a modern aircraft, which presents problems in geometry grid generation, fluid flow, acoustics, structural analysis, operational research, visualization, and database management. The Metacomputer for such a Metaproblem would be networked workstations, array processors, vector supercomputers, massively parallel processors, and visualization engines.

Multimedia Server or Client Multimedia refers to information (digital data) with different modalities, including text, images, video, and computer generated simulation. Servers dispense this data, and clients receive it. Some form of browsing, or searching, establishes which data is to be transferred. See also *InfoVISiON*.

Multiple-Instruction/Multiple-Data (MIMD) A parallel computer architecture where the nodes have separate instruction streams that can address separate memory locations on each clock cycle. All *HPDC* systems of interest are *MIMD* when viewed as a *metacomputer*, although the nodes of this metacomputer could have *SIMD* architectures.

Multipurpose Internet Mail Extension (MIME) The format used in sending multimedia messages between *Web Clients* and *Servers* that is borrowed from that defined for electronic mail.

National Information Infrastructure (NII) The collection of *ATM*, cable, *ISDN*, *POTS*, satellite, and wireless networks connecting the collection of 10^8–10^9 computers that will be deployed across the U.S.A. as set-top boxes, PCs, workstations, and MPPs in the future.

The NII can be viewed as just the network infrastructure or the full collection of networks, computers, and overlayed software services. The *Internet* and *World Wide Web* are a prototype of the NII.

Network A physical communication medium. A network may consist of one or more buses, a switch, or the links joining processors in a multicomputer.

Node A parallel or distributed system is made of a bunch of nodes or fundamental computing units—typically fully fledged computers in the *MIMD* architecture.

N(UMA) UMA—Uniform Memory Access—refers to shared memory in which all locations have the same access characteristics, including the same access time. NUMA (Non-Uniform Memory Access) refers to the opposite scenario.

Parallel Computer A computer in which several functional units are executing independently. The architecture can vary from *SMP* to *MPP* and the *nodes* (functional units) are *tightly coupled.*

POTS The conventional twisted pair based Plain Old Telephone Service.

Protocol A set of conventions and implementation methodologies defining the communication between nodes on a network. There is a famous seven layer OSI standard model going from physical link (optical fiber to satellite) to application layer (such as Fortran subroutine calls).

Any given system, such as PVM or the Web implements a particular set of protocols.

PVM PVM was developed at Emory and Tennessee Universities, and Oak Ridge National Laboratory. It supports the *message passing* programming model on a network of heterogeneous computers (http://www.epm.ornl.gov/pvm/).

Shared Memory Memory that appears to the user to be contained in a single address space that can be accessed by any process or any node (functional unit) of the computer. Shared memory may have *UMA* or *NUMA* structure. *Distributed computers* can have a shared memory model implemented in either hardware or software—this would always be NUMA. Shared memory parallel computers can be either *NUMA* or *UMA*.

Virtual or Distributed Shared Memory is (the illusion of) a shared memory built with physically distributed memory.

Single-Instruction/Multiple-Data (SIMD) A *parallel computer* architecture in which every node runs in lockstep accessing a single global instruction stream, but with different memory locations addressed by each node. Such synchronous operation is very unnatural for the nodes of a HPDC system.

Supercomputer the most powerful computer that is available at any given time. As performance is roughly proportional to cost, this is not very well defined for a scalable parallel computer. Traditionally, computers costing some $10–$30 M are termed supercomputers.

Symmetric Multiprocessor (SMP) A Symmetric Multiprocessor supports a shared memory programming model—typically with a *UMA* memory system, and a collection of up to 32 nodes connected with a bus.

Televirtual The ultimate computer illusion where the user is fully integrated into a simulated environment and so can interact naturally with fellow users distributed around the globe.

Teraflop A measure of computer performance—one Teraflop is 10^{12} floating point operations per second.

Transmission Control Protocol (TCP) A connection-oriented transport protocol used in the DARPA Internet. TCP provides for the reliable transfer of data, as well as the out-of-band indication of urgent data.

VBNS A high speed *ATM* experimental *network (WAN)* maintained by the National Science Foundation (NSF) to link its four *Supercomputer* centers at Cornell, Illinois, Pittsburgh, and San Diego, as well as the Boulder National Center for Atmospheric Research (NCAR).

Virtual Reality Modeling Language (VRML) A "three-dimensional" HTML that can be used to give a universal description of three-dimensional objects that supports *hyperlinks* to additional information.

Web Clients and Servers A distributed set of clients (requesters and receivers of services) and servers (receiving and satisfying requests from clients) using *Web Technologies*.

WebWindows The operating environment created on the World Wide Web to manage a distributed set of networked computers. WebWindows is built from *Web clients* and *Web servers*.

WebWork (Fox:95a) An environment proposed by Boston University, Cooperating Systems Corporation, and Syracuse University, which integrates computing and information services to support a rich distributed programming environment.

World Wide Web and Web Technologies A very important software model for accessing information on the Internet based on hyperlinks supported by *Web technologies*, such as *HTTP*, *HTML*, *MIME*, *Java*, *Applets*, and *VRML*.

References

[Almasi:94a] Almasi, G. S., and Gottlieb, A. *Highly Parallel Computing.* The Benjamin/Cummings Publishing Company, Inc., Redwood City, CA, 1994. second edition.

[Andrews:91a] Andrews, G. R. *Concurrent Programming: Principles and Practice.* The Benjamin/Cummings Publishing Company, Inc., Redwood City, CA, 1991.

[Angus:90a] Angus, I. G., Fox, G. C., Kim, J. S., and Walker, D. W. *Solving Problems on Concurrent Processors: Software for Concurrent Processors*, volume 2. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1990.

[Birman:92a] Birman, K., et al. *ISIS User Guide and Reference Manual.* Isis Distributed Syustems, Inc., Ithaca, NY, 1992.

[Foster:95a] Foster, I. *Designing and Building Parallel Programs.* Addison-Wesley, 1995. http://www.mcs.acl.gov/dbpp/.

[Fox:88a] Fox, G. C., Johnson, M. A., Lyzenga, G. A., Otto, S. W., Salmon, J. K., and Walker, D. W. *Solving Problems on Concurrent Processors*, volume 1. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1988.

[Fox:89y] Fox, G. C. "Parallel computing." Technical Report C3P-830, California Institute of Technology, Pasadena, CA, September 1989. Chapter in *Encyclopedia of Physical Science and Technology 1991 Yearbook*, Academic Press, Inc.

[Fox:90o] Fox, G. C. "Applications of parallel supercomputers: Scientific results and computer science lessons," in M. A. Arbib and J. A. Robinson, editors, *Natural and Artificial Parallel Computation*, chapter 4, pages 47–90. MIT Press, Cambridge, MA, 1990. SCCS-23. Caltech Report C3P-806b.

[Fox:92b] Fox, G. C. "Parallel supercomputers," in C. H. Chen, editor, *Computer Engineering Handbook*, chapter 17. McGraw-Hill Publishing Company, New York, 1992. Caltech Report C3P-451d.

[Fox:92c] Fox, G. C. "The use of physics concepts in computation," in B. A. Huberman, editor, *Computation: The Micro and the Macro View*, chapter 3, pages 103–154. World Scientific Publishing Co. Ltd., 1992. SCCS-237, CRPC-TR92198. Caltech Report C3P-974.

[Fox:94a] Fox, G. C., Messina, P. C., and Williams, R. D., editors. *Parallel Computing Works!* Morgan Kaufmann Publishers, San Francisco, CA, 1994. http://www.infomall.org/npac/pcw/.

[Fox:95a] Fox, G. C., Furmanski, W., Chen, M., Rebbi, C., and Cowie, J. H. "WebWork: integrated programming environment tools for national and grand challenges." Technical Report SCCS-715, Syracuse University, NPAC, Syracuse, NY, June 1995. Joint Boston-CSC-NPAC Project Plan to Develop WebWork.

[Fox:95c] Fox, G. C. "Software and hardware requirements for some applications of parallel computing to industrial problems." Technical Report SCCS-717, Syracuse University, NPAC, Syracuse, NY, June 1995. Submitted to ICASE for publication (6/22/95); revised SCCS.134c.

[Fox:95d] Fox, G. C., and Furmanski, W. "The use of the national information infrastructure and high performance computers in industry," in *Proceedings of the Second International Conference on Massively Parallel Processing using Optical Interconnections*, pages 298–312, Los Alamitos, CA, October 1995. IEEE Computer Society Press. Syracuse University Technical Report SCCS-732.

[Fox:95e] Fox, G. C. "Basic issues and current status of parallel computing." Technical Report SCCS-736, Syracuse University, NPAC, Syracuse, NY, November 1995.

[Hariri:95a] Hariri, S., and Lu, B. *ATM-based High Performance Distributed Computing.* McGraw Hill, 1995. Zomaya, Albert (editor).

[Hariri:96a] Hariri, S. *High Performance Distributed Computing: Network, Architecture and Programming.* Prentice Hall, 1996. To be published.

[Hillis:85a] Hillis, W. D. *The Connection Machine.* MIT Press, Cambridge, MA, 1985.

[Hockney:81b] Hockney, R. W., and Jesshope, C. R. *Parallel Computers.* Adam Hilger, Ltd., Bristol, Great Britain, 1981.

[HPCC:96a] National Science and Technology Council, "High performance computing and communications," 1996. 1996 Federal Blue Book. A report by the Committee on Information and Communications. Web address http://www.hpcc.gov/blue96/.

[Kung:92a] Kung, H. T. "Gigabit local area networks: A systems perspective," *IEEE Communications Magazine*, April 1992.

[McBryan:94a] McBryan, O. "An overview of message passing environments," *Parallel Computing*, 20(4):417–444, 1994.

[Morse:94a] Morse, H. S. *Practical Parallel Computing.* Academic Press, Cambridge, Massachusetts, 1994.

[Narayan:95a] Narayan, S., Hwang, Y., Kodkani, N., Park, S., Yu, H., and Hariri, S. "ISDN: an efficient network technology to deliver information services," in *International Conference on Computer Application in Industry and Engineering*, 1995.

[Nelson:90b] Nelson, M. E., Furmanski, W., and Bower, J. M. "Brain maps and parallel computers," *Trends Neurosci.*, 10:403–408, 1990.

[Pfister:95a] Pfister, G. F. *In Search of Clusters: The Coming Battle in Lowly Parallel Computing.* Prentice Hall, 1995.

[Stone:91a] Stone, H. S., and Cocke, J. "Computer architecture in the 1990s," *IEEE Computer*, pages 30–38, 1991.

[SuperC:91a] *Proceedings of Supercomputing '91*, Los Alamitos, California, 1991. IEEE Computer Society Press.

[SuperC:92a] *Proceedings of Supercomputing '92*, Los Alamitos, California, November 1992. IEEE Computer Society Press. Held in Minneapolis, Minnesota.

[SuperC:93a] *Proceedings of Supercomputing '93*, Los Alamitos, California, November 1993. IEEE Computer Society Press. Held in Portland, Oregon.

[SuperC:94a] *Proceedings of Supercomputing '94*, Los Alamitos, California, November 1994. IEEE Computer Society Press. Held in Washington, D.C.

[Tanenbaum:95a] Tanenbaum, A. S. *Distributed Operating Systems.* Prentice Hall, 1995.

[Trew:91a] Trew, A., and Wilson, G. *Past, Present, Parallel: A Survey of Available Parallel Computing Systems.* Springer-Verlag, Berlin, 1991.