LBL-31610
UC-905

Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

Information and Computing Sciences Division

Information Technology Resources Assessment

D.F. Stevens, Editor

January 1992

Prepared for the U.S. Department of Energy under Contract Number DE-AC03-76SF00098

LBL--31610

DE92 009498

Information Technology Resources Assessment*

David F. Stevens, Editor
Information and Computing Sciences Division
Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

January, 1992

Prepared for the
Department of Energy Information Resources Management
Long-Range Plan: FY 1993 - FY 1997

* This work was supported by the U. S. Department of Energy under contract No. DE-AC03-76SF00098.

Information Technology Resources Assessment*

1. Introduction

The 1992 ITRA places more emphasis on the expected effects of the use of new IT within DOE than simply on expected advances of the underlying technology.

This year's Information Technology Resources Assessment (ITRA) is something of a departure from traditional practice. Past assessments have concentrated on developments in fundamental technology, particularly with respect to hardware. They form an impressive chronicle of decreasing cycle times, increasing densities, decreasing costs (or, equivalently, increasing capacity and capability per dollar spent), and new system architectures, with a leavening of operating systems and languages. Past assessments have aimed—and succeeded—at putting information technology squarely in the spotlight; by contrast, in the first part of this assessment, we would like to move it to the background, and encourage the reader to reflect less on the continuing technological miracles of miniaturization in space and time and more on the second- and third-order implications of some possible workplace applications of these miracles.

This is a particularly opportune time for this approach, because the way has been prepared by the publication in September, 1991, of two special magazine issues that address our future as users of information technology (IT): "The Promise of the Next Decade" (*IEEE Computer*) and "Communications, Computers, and Networks" (*Scientific American*). It is also more in tune with the thrust of the DOE IRM Vision 21 objectives of service, institutionalization, and standardization, as well as technology.

This Information Technology Resources Assessment is intended to provide a sense of technological direction for planners in projecting the hardware, software, and human resources necessary to support the diverse IT requirements of the various components of the DOE community. It is also intended to provide a sense of our new understanding of the place of IT in our organizations.

* This work was supported by the U. S. Department of Energy under contract No. DE-AC03-76SF00098. It was prepared as Appendix B of the *Information Resources Management Long-Range Plan: FY 1993 - FY 1997.*

Figure 1 illustrates this as a kind of phase diagram, showing on the left the 80's vision of IT, dominated by technology, and on the right, the balanced emphasis set forth in Vision 21. Along the bottom we have indicated the approximate position in this phase space of some of the developments mentioned in this assessment. The two rightmost examples have been chosen to illustrate that Vision 21 does not apply only to new applications (such as technologically-augmented instruction, but also and equally to extensions of familiar applications (such as intelligent document processing).

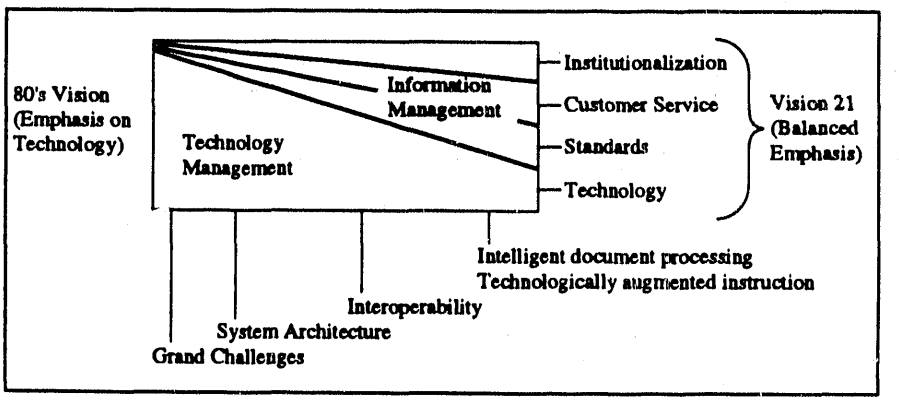

Figure 1
Toward Vision 21

2. The Coming Invisibility of IT

2.1. On Technological Revolutions

IT has been called a revolutionary technology; to realize its revolutionary potential, a technology must become suitably integrated into the mental model of the workers as well as into the work itself. Integration into the work is a three-stage process: faster processing, new methods, and new work.

Information technology, and computing in particular, has often been declared to be a revolutionary technology, with no further discussion of what it might mean to be "revolutionary". There are at least two viewpoints from which to examine a technology to determine if it qualifies as revolutionary: the integration of the technology into the work, and its integration into the mental model of the workers. Integration of a revolutionary technology into the work is a three-stage process of acceleration, adaptation, and expansion. In the first stage, we do the same work in the same way, but faster. In the second, we continue to do the same work, but in dramatically new ways. The third stage is the introduction of wholly new work: As work is done differently, new kinds of work will come to be done.

Document preparation provides an excellent example. In the first stage, IT was applied to speeding up the process of preparing traditional documents in traditional ways, i.e., the originator of the material turned preliminary copy over to an array of specialists (secretaries, editors, etc.) who then used the new tools to produce better-looking documents faster. In the second stage (where much of DOE now works), we still prepare the same sorts of documents, but much of the work is now done directly by the originator, using the new tools to do the work unaided. We have not moved into the third stage, but one possible scenario would be the introduction of hypertextual material, in which traditional documents, which consist of serially accessible sets of sentences, are replaced by hyperdocuments, consisting of multiply-linked masses of material in several media.

The four stages of integration: the priesthood, the guild, the conscious use of tools, and the unconscious use of tools.

Integration of IT into the mental model of the workforce is nearing the end of a four-stage process: mediation through priests, execution by expert craftsmen, concentration on the tools, and concentration on the work. In the first two stages, the people for whom the work was to be done were ill-equipped to do it themselves. Instead

they had to hand it over to specialists. In the first stage, all useful work was accomplished out of sight; the actual beneficiary usually didn't talk directly to the designers and programmers, but only to special intermediaries (systems analysts), who were trained both in the specialized artificial languages of the programmers' craft and (often less successfully) in the the ordinary languages of the users.

As the mystery waned, but the need for detailed technical knowledge and skill remained, users were allowed to speak directly to the experts who translated their desires into working programs, but were still not allowed direct access to the machines themselves. The third stage, in which most of us now reside, is characterized by the existence of a wealth of sophisticated and powerful tools that can be wielded by the ordinary person, but only with a conscious effort, and after a preliminary phase of learning an elementary technical vocabulary, the rudiments of system structure, and the tricks necessary to achieve the particular effects we have in mind (such as the formatting of this section). In the fourth stage, which will soon be upon us, the workforce will have grown up with a large assortment of sophisticated tools and techniques, whose use will be second nature to them as the use of electricity, telephone, and internal combustion is to us; they will be able to concentrate on the work instead of the tools.

Successful technology does not guarantee a successful technological revolution; it must be supported by universal accessibility, a new infrastructure, and a new mind-set.

We have learned from earlier technological revolutions (e.g., ground transportation, communications) that the mere ability to revolutionize our work life is not enough to bring about a revolution; for the revolution to actually take place, three additional elements are necessary:

- the technology must be accessible to everyone (in the dual sense that it is both available to all and requires no unusual dexterity, talent, or strength to operate);
- it must have inspired the development of a ubiquitous supporting infrastructure; and
- it must have been around long enough for the current generation to see it as part of the background rather than as part of a revolution.

Although work of a revolutionary nature *can* be accomplished in a limited environment without universal accessibility, ubiquitous infrastructure, and universal familiarity, it remains experimental and prototypical without them. In their absence, the revolution will not spread. The most obvious of these requirements is accessibility: A revolutionary technology that is available to only a few is unlikely to change the lives of the many. But even if the technology is widely available, it may have only limited influence if a suitable supporting infrastructure has not been developed, and its potential will not be fully tapped until the population as a whole takes it for granted. In the case of ground transportation, railroads began but could not complete the revolution: they were never available in a fully useful way to everyone, and they provided only marginal service to those who were not actually on a rail line. Completion was provided by the automobile, but not until after self-starters and synchromesh transmissions provided universal accessibility; high-performance fuels, paved roads, and gas stations provided a supporting infrastructure; and the passage of 40 years provided a population that had never known life without automobiles.

IT is in everyone's workplace, an infrastructure of interconnectivity is coming into being, and the next generation of workers will treat high-performance IT tools as the norm.

IT is now reaching the stage of full integration into American life. Computers are more accessible already than automobiles; the continuing march of the technology itself—which shows no sign of slowing—has allowed small versions to be incorporated into giveaway and throwaway items. Computers are incorporated into almost every tool of modern life, from jetliners to telephones. An infrastructure of interconnectivity now exists, and interoperability is in every vendor's sales vocabulary, even those for whom such a concept was anathema as recently as three years ago. The advances in usability that accompanied—and enabled—the rise of the personal, computer-assisted workstations have been enormous, making working computers almost as easy to use as those found in a video arcade. The presence of computers in cars, phones, schools, games, toys, hotel lobbies, etc., means that our children and grandchildren are growing up with no awareness of computers as marvels, but simply

as things that are generally around. The technology itself is becoming invisible, and the users of the technology are becoming ever more free to focus their full attention on the goal rather than the process.

The next generation of workers will be far less interested in the speed with which IT tools can do dull things than in the ease with which they can do exciting things, so they will develop exciting applications to make use of the tools. We cannot yet predict what those exciting things will be or how they will change the way we work, but we can see some emerging developments that have interesting possibilities.

Examples include hypermedia, computer-supported cooperative work, and artificial reality.

Three of these developments are hypermedia, groupware or computer-supported cooperative work, and artificial/ virtual reality. In the balance of this section, we shall consider them in more detail, with an eye towards predicting how they may be expected to change the nature of work within DOE; we shall then look briefly at technologically-augmented instruction as a single integrated application of these new technologies, and at intelligent document processing as a prototypical example of a Vision 21 approach to an existing application. Finally, we shall address the implications of these changes for institutional ITR architectures. In the last section, we shall take a more traditional look at selected technology.

2.2. Hypermedia

Documents as multimedia presentations with content-driven links to related matter.

The development of hypermedia has been waiting to happen for some years—or even decades* —but it has not had the necessary supporting infrastructure. A hyperdocument is a multidimensional generalization of a conventional document. A traditional document consists primarily of linear text, possibly accompanied by tabular and (static) pictorial material, and references (pointers) to related material, written on paper. In the personal computing environment, we have extended the concept to include information contained on other media, but still or-

* It is strongly related to the *memex* concept proposed by Vannevar Bush in 1945; the term *hypertext* itself has been in use since at least 1965.

ganized for the most part in a linear or tabular fashion. A hyperdocument embodies a further extension, to include animated images and video clips as well as static pictorial material; open links to referenced material, possibly including appropriate recordings or real-time satellite views, as well as pointers; version histories and variant readings available at the touch of a button; real or synthesized auditory information—any information, in short, that is digitizable—organized in network fashion, so that navigation can follow the vagaries of the reader's thought rather than being constrained to a linear structure imposed by the writer.

Several pieces of the applicable technology are available today (linked spreadsheets, sound effects attached to screen utilities, and electronic marginalia already exist, for example, and a language (Hytime, based on the existing Standard Generalized Markup Language) for processing hyperdocuments is on the verge of international standardization), but the whole package hasn't quite been put together in an integrated, transportable form. Nevertheless, a few innovative applications have appeared. In evaluating progress in this area, however, we need to be wary of defining the term *hyperdocument* too narrowly. An adventure game is a hyperdocument involving text, animated video, and sound, with the sequence of effects being determined in real time by the action of the player. A more dramatic application is a Stephen Hawking lecture: Dr. Hawking suffers from amyotrophic lateral sclerosis (ALS) and cannot speak; his "lecture" is an audio hyperdocument created with the aid of a reduced keyboard and a voice synthesizer.

2.3 Groupware

With groupware, we are replacing the metaphor of the desktop with the metaphor of the conference room or lecture hall.

We are now also beginning to see some startling possibilities in the area of computer-supported cooperative work (CSCW), or groupware. Earlier developments have included various forms of computer-assisted conferencing, messaging, and e-mail; telephonic and video teleconferencing; facsimile transmission; and even the photocopier. They have all been first- or early-second-stage de-

velopments applied to reducing the physical limitations we now experience in collaborative work. To achieve real-time interaction, most of us still think in terms of gathering in the same place at the same time. Voice mail and e-mail allow us to converse asynchronously and, in the latter instance, with many people at once. The various forms of teleconferencing allow us to replace human travel with network connectivity. The photocopier and facsimile machine eliminate the delays formerly inherent in the distribution of multiple copies of joint documents. Two distinct threads are beginning to emerge from this work: a shift in the dominant user-interface metaphor and a rôle-reversal of sorts between an application and the platform on which it is run.

The dominant interface metaphor in the workplace today is that of the desktop, in which the "objects" with which one works are designed to emulate certain of the properties of such familiar objects as memos, letters, file folders, and the like. The dominant groupware metaphor will be based on the room (or even the building) rather than the desktop. The room might be a university lecture hall or a conference room. In these cases, the objects with which one will work might include electronic chalkboards, flipcharts, and slide and overhead projections; it will be possible to interact directly with the instructor, or with a subgroup, to share a single (hyper)document or to work on simultaneous versions, to enjoy instantaneous translation, etc. The desktop metaphor will not disappear, for one will wish to retain the ability to retire to one's own "office" from a lengthy conference, but as telecommuting replaces physical commuting, the ad-hoc conference room will come more and more to dominate our electronic work.

We may also see the advent of the nomadic application, which follows its master from place to place.

The other thread is a shift in emphasis away from today's version of distributed computing, in which the central intelligence of an application remains fixed at one workstation, but pieces are farmed out to specific remote computers, to what might be called nomadic applications, which follow their originator about the building (or the world) using whatever computing resources come to hand. We will see more and more applications designed for people who are away from their desks.

There are likely to be significant organizational impacts.

The IT research community has adopted CSCW as one of the waves of the future. A conference on the topic, jointly sponsored by the ACM special-interest groups on computers and human interaction (SIGCHI) and office inforformation systems (SIGOIS), took place in October of 1990. One of the the topics to receive significant attention was the effect of CSCW on the organization. Already we can see hints of the changes that will follow in the wake of the development of effective groupware. Experience with e-mail, for example, has shown that the introduction of facelessness and the elimination of visual cultural cues can change the nature of interpersonal interactions, making it far more difficult to convey delicate shades of meaning. Shoshana Zuboff* notes that the spread of these technologies is changing the way in which information flows through an organization, and that existing organizational structure may not be well-suited to the coming information dynamism.

2.4. Virtual Reality

Virtual reality allows a scientist to take a walk through a reactor going critical, or to create new molecules atom-by-atom.

Groupware and hypermedia are still primarily concerned with work as we know it today. By contrast, artificial or virtual reality provides a hint of the startling possibilities that exist for future work. Simplistically, virtual reality can do for an investigator what Industrial Light and Magic does for George Lucas, namely, create a whole world out of one's imagination. Initially, this sounds like frivolity run rampant. In practice, however, we can expect virtual reality to promote our understanding of many complex and mysterious processes.

Current computational analysis allows DOE researchers to simulate experiments that are undesirable or impossible to perform physically, and to cope with natural phenomena that are too big, too small, too fast, or too slow for direct human observation. Frequent physical testing of atomic devices is undesirable. An ocean is too big to see and understand. A molecule is too small, the behavior of a fluid is too fast, the birth of a galaxy too slow. Computational science allows scientists to model and visualize these types of phenomena in a way they can

* *In the Age of the Smart Machine: The Future of Work and Power*, Basic Books, 1989.

study, but current techniques are still dominated by the generation, display, and interpretation of numbers and curves. More evocative ways of presenting information are beginning to appear, so far involving primarily color and motion. As we move toward true virtual reality, the full range of senses can be brought into play.

Virtual reality will allow the investigator to replace numerical analysis with physical experience. He will be able to see events happening in the far ultraviolet or infrared, or at speeds and sizes beyond the resolving capacity of the human eye; to stand in the fuel intake of an automotive cylinder during operation, observe the formation of pollutants, and change the molecular composition of the fuel on the spot to test theories of pollution reduction; to hear the song of an earthquake precursor; to feel the resistance of living tissue as he practices microsurgery.

2.5. Technologically-Augmented Instruction

Hypermedia and groupware in the distributed classroom.

A single application area in which all three of these emerging developments will be involved is education, with the creation of what might be called *technologically--augmented instruction*. Technologically-augmented instruction will combine the distributed lecture-hall-*cum*-classroom, supported by groupware, with multiply-accessible hyperdocuments and virtual experimental entities (creatures, environments, mathematical constructs, even universes, if need be) to imbue instruction with an immediacy that is difficult to imagine if we draw on our own past experience as students. Through technological innovations, instructional materials that are now considered to be single-instance (such a s the instructor, a human heart with a rarely-occurring defect, a great work of art or architecture) can be made available simultaneously to students in many locations. In many of these cases (the instructor being an obvious exception), the students can experiment freely with the materials. It is possible, for example, that the basic text "document" in a biology course will be a virtual live frog whose circulatory system the student can (non-destructively) explore as if he were a self-propelled blood-cell, or that

narrative history is replaced with apparently live observation of significant events.

2.6. Intelligent Document Processing

Dealing with the real reality.

While virtual reality is the glamorous side of the IT-based working world, some of us will have to continue to deal with the *real* reality: the creation, distribution, digestion, reconstitution, and regurgitation of vast amounts of technical and administrative information. Today, that means dealing with vast amounts of paper; tomorrow, it may mean something quite different.

The large organizations of today have grown up around the concept of paper flow. They are designed to create paper efficiently, route it to those who need to know the contents, and store it in a reasonably recoverable fashion. The first of these goals has far outstripped the other two: Our organizations have so many layers that routing of one version of a document often lags behind creation of the next, and the quantity has become so great that storage and retrieval are increasingly difficult.

Paper-based information flow leads to information hoarding, excess locality, distribution by status instead of by need-to-know, and inflexibility in handling information.

The limitations of paper have led to a particular style of information management, characterized by information hoarding, locality, and inflexibility. Paper is a physical medium, and our management of the information it contains is often subverted by the procedures we have adopted for controlling the medium. Because the making of copies has been historically difficult, their possession was a matter of some importance. This has led rather naturally to a tendency to treat information as a local asset to be guarded zealously, with the result that information is hoarded. Distribution becomes a matter of status, and is based on one's position in the organization instead of one's need-to-know. Because information is a local asset, it is viewed only from a local perspective, and changes in format and content are imposed for local reasons. Storage is based on predefined indexing schemes—that often vary from office to office within an organization—so that retrieval across organizational boundaries becomes a major mystery. And the content is fixed once the paper is typed/ printed, so that alternate views of the data are extremely difficult to generate.

Electronic-based information handling encourages the treatment of data as a corporate asset, allows content-based distribution and handling, and facilitiates the tailoring of information to the needs of the moment.

This contrasts rather starkly with the sort of information management style that will become possible in the mid-90's. Information is coming to be seen as a corporate asset, and the technology to support that view is in sight. The limitations of print on paper are being overcome as tools become generally available than can handle any recordable medium. As immediate capture of information becomes possible, the number of filters will diminish. As corporate-wide access becomes possible, the hoarding of information will be discouraged. As content-based treatment becomes possible, distribution will reflect relevancy rather than organizational position. At the same time, the ability to tailor extracts dynamically to the needs of the local unit, and to combine data from different, formerly incompatible databases, will allow local variants to be created on the fly, kept up-to-date automatically, and discarded when no longer useful.

Why choose document processing as an example?

It may have come as something of a surprise that we have chosen document processing as one of our prototypical Vision 21 applications, because for many of us, document processing is our most intimate connection with IT. We have seen marvelous things come to pass in the last few years, and tend to think of document processing as a rather mature application. But these past developments have had a strong technological flavor: cheap cycles and storage, high-quality printing, improved user interfaces. As the other emphases of Vision 21—interoperability (the result of the adoption of standards), customer service orientation, and institutionalization—come into play we shall begin to see intelligent document processing supplant rapid document preparation as the dominant theme in this area.

The basic elements of intelligent document processing have been characterized as getting the right information to the right people in the right form at the right time, all at a reasonable cost. Technology alone does not address all of these elements. Interoperability is necessary to allow the amalgamation of information created in many places and stored in many formats. A customer service orientation means that the system will tailor the information to the needs of the user, instead of forcing the

user to change the statement of his requirements to fit the constraints of the system. Institutionalization is the organizational analogue of individual integration of IT capabilities into one's mental model. It implies acceptance of IT as a utility as fundamental as electricity, rather than its traditional treatment as a controlled substance.

Adding intelligence to documents, storage systems, and distribution systems.

To appreciate fully the potential of intelligent document processing, it is necessary to reflect on the location of the intelligence involved. Today, that intelligence is generally limited to the originators and creators of the document concerned. It is they who decide content, format (within the constraints of their document preparation systems), and distribution. Those who could benefit from the content, but who are not on the distribution list often have no way of discovering the existence of the document, much less of obtaining access to it. More interesting possibilities are surfacing. Intelligence can reside in storage and distribution systems, and even in the document itself. Thus it will soon be possible to create a single "library" consisting of all the document repositories of, for example, all the DOE laboratories.

It is already possible to browse the physics preprint catalogue maintained by the Stanford Linear Accelerator Center (SLAC) from a remote location; if the preprints themselves had been created in machine-readable form, it would soon be possible to read the whole text. The next step is automatic notification to the user when a paper of possible interest has been added to the collection; this has already been implemented in prototype form as part of the Apple Corporation Navigator project.

2.7. Implications for Institutional ITR Architectures

2.7.1. Environments

Institutional ITR architectures must change as the technology and the work changes. The current notion of what constitutes a "standard" architecture is changing from being platform driven to being environment driven.

Institutional ITR architectures are a function of the technology we have available, the work we have to do, and our attempts to bring the power of the one to bear on the problems of the other. Institutional ITR architectures change in response to changes in these parameters. In the last 25 years, the ITR architectural "standard" has shifted from a monolithic central system to a confederation of semi-independent, semi-detached systems, and thence to a myriad of interconnected individual systems, but to a large extent, it can still be characterized as *platform driven*. The next few years will see a shift to *environment-driven* architectures, consisting of specialized working environments built upon an increasingly common set of mechanisms and baseline technologies. (See Figure 2.) Three environments that are coming into prominence are collaborative work, on-line information, and real-time data collection. It is not the existence of such work environments that is new, but the expectation that they will be part of the everyday work environment for increasing numbers of DOE employees and contractors.

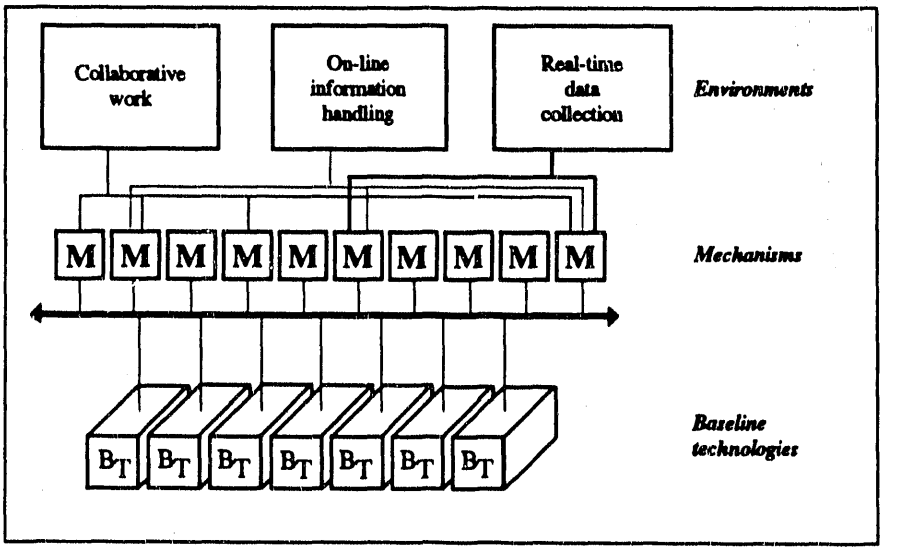

Figure 2
User Environments

Collaborative work

Groupware will be tailored for scientific applications.

We have already spoken of groupware as one of the major current developmental thrusts. It will have particular application in the DOE National laboratories as the operative environment for the scientist becomes dominated by computer-assisted collaborative work. This environment will be supported by specialized tools tailored to provide assistance to scientists in such areas as

- collaborative presentations;
- joint authorship of publication materials;
- collaborative operation of experiments; and
- collaborative code development.

Such collaborations already exist; only the augmented technological infrastructure is lacking.

On-line information handling

On-line information will increase in breadth of information available and in breadth of access.

Much of the work conducted by DOE consists of the distribution and analysis of information. As more and more of this information is stored in digital, machine-readable form, the demand for on-line information handling will grow in several distinct areas. Examples of on-line information-handling environments that can be expected to develop include

- scientific database environments, containing
  - -- archived results of simulations;
  - -- experimental information (calibration, raw data);
  - -- results (e.g., Genbank).
- library environments, either containing or (more likely) providing access to
  - -- standard reference materials currently found in hardcopy form;
  - -- indexes and "current contents" for most published material;
  - -- physical and chemical properties tables;
  - -- engineering handbooks;
  - -- drawings, maps, and photographs;
  - -- video and audio reference materials.
- corporate database environments that provide controlled access to

-- personnel information;
-- budget and accounting information;
-- stores and purchasing information;
-- proposals and documents;
-- management communications.

Real-time data collection

Even though real-time data collection is extremely context-dependent, it will use standard mechansims.

Some areas of DOE, such as high-energy physics, are already deeply dependent upon extremely sophisticated real-time data collection environments. These environments are often founded on reduced versions of standard operating systems, specialized for high performance, augmented with hardware and software to interface with experimental apparatus. Although the experimental interface will employ specially-designed mechanisms, they will be interconnected with discipline-standard processing modules that make heavy use of standard mechanisms and baseline technologies.

2.7.2. Mechanisms

"Mechanisms" are baseline technologies that have been accepted as general-purpose tools.

The dividing line between "mechanism" and "baseline technology" constantly shifts as new technology achieves the reliability and interoperability to be incorporated into the baseline, and older technology is packaged into the building-block mechanisms that support the standard computing culture. X-windows is an example of technology that has recently made the transition to baseline technology—today one would expect support for X-windows to be included in any off-the-shelf Unix system. Database management systems may soon make a similar transition (if the price comes down).

Examples of other technologies that are gaining acceptance as mechanisms are

- application specific tool modules;
- integrated "electronic publishing" systems that support graphics, video, text, audio, and multiply-interconnected document organization (hypertext);
- iconic programming environments;
- window application development tools.

2.7.3. Baseline Technologies

The baseline technologies are those upon which all the mechanisms depend and through which they communicate. A few of them are the subject of later sections of this Assessment. Some of the others that bear watching are listed below. (They have all been addressed in recent DOE Information Technology Resources Assessments.)

- POSIX OS and runtime environment
- security and access control systems
- X-window support (although this would benefit from better consensus on style)
- interprocess communication standards
- standard network protocols
  - -- RPC (remote procedure call) and data presentation
  - -- sockets
  - -- TCP/IP (the standard Internet protocol)
- mass storage systems
- high speed (>100 megabits per second) networks (the National Research and Education Network (NREN), for example)
- user workstation capabilities
  - -- high resolution color
  - -- audio, video input and output
  - -- mouse and multidimensional pointing/positioning devices
  - -- > 128 megabytes of memory
  - -- > 100 MIPS of processing power (a MIPS is one million instructions per second)

2.8. The Desktop Environment

The desktop as the hub of an ad hoc *computer.*

We anticipate that software, computing hardware, and communications advances over the next few years will result in an information analysis environment in which scientists will have uniform and unimpeded access to computing and data resources regardless of their geographic location. Within this environment a scientist will be able to assemble, at a moment's notice, a set of resources suitable to the needs of the problem under consideration, use this tailor-made *ad hoc* computer as

necessary, and then return the resources to the common pool. The flexibility and power inherent in such a scheme will qualitatively improve our ability to address the difficult and large-scale problems of the Grand Challenges addressed by the High Performance Computing and Communications program (HPCC). This dramatic development will be enabled by advances in several areas. It is worth noting that four of these correspond rather directly to the fourfold emphasis of Vision 21, while the remaining one embodies all four.

- Improvements in workstation performance and architecture will give us an order of magnitude increase in the critical areas of I/O capacity and the routine incorporation of co-processors for encryption and video compression/decompression. (Technology.)
- The national commitment to widely available, high bandwidth networking through NREN and advances in network protocols and security architectures will move distributed access to resources and data into the mainstream. (Institutionalization.)
- Hardware and software improvements will permit multiple heterogeneous computing systems to be easily configured to cooperate routinely on diverse problems. (Standardization and interoperability.)
- Easy access to massive unique data archives will be enabled through advances in data management and mass storage systems. (All four.)
- User interface paradigms will evolve that provide non-computer specialists with straightforward mechanisms for the assembly of the above elements into effective tools to attack scientific problems. (Customer service orientation.)

A richer desktop environment.

These changes will lead directly to a desktop environment that is substantially richer and easier to use than any currently available computing situation. They will provide an enormous degree of interconnection that permits easy and flexible experiment design and execution.

- *desktop video*

  Video compression based on faster processors and increased storage will provide multimedia teleconferencing; storage, retrieval, and manipulation of video sequences of scientific experiments; and development of much more sophisticated image indexing and query systems. This will permit quick and easy review of project history, study, and integration with other related information.

- *temporary problem-specific configurations utilizing external resources*

  Advances in interprocess communication and the underlying networking technology will permit design of architectures for solving specific problems to be implemented by temporarily interconnecting the optimal computing elements. This removes geographic restrictions and allows easy sharing of expansion and scarce elements (e.g., specialized processors). For example, high speed data collection systems will write data over the course of an experiment into a high speed data buffer like a large network RAID (redundant array of inexpensive disks) system. Data from this "buffer" can be accessed by many different processes to accomplish immediate analysis for control of the experimental device (the data generator), organization and filtering for more complete integrated analysis in the context of other experiments, and filtering and compression for long term storage.

- *and access to the whole collaboration*

  This logical configuration may consist of large, expensive elements (several types of supercomputer, a large RAID system, an on-line massstorage system, etc.) and may involve many people (engineers involved in experiment control, collaborators previewing data, and students observing the whole process) all of whom participate via network-based workstations. The physical configuration as represented by the data flows between the elements and participants will last only the few seconds, minutes, hours, days, of the actual experiment.

  When the experiment cycle is complete the resources are released for other use. The resources—computing and storage elements—may also be shared during the course of the experiment depending on capacity requirements. The software and hardware being developed all have shared use as a design criterion.

3. Scientific Computing

3.1. The Grand Challenges

3.1.1 Introduction

Some problems demand computational power beyond our present capability.

Although we now recognize that *information technology* encompasses more than just raw computational power, we continue to face fundamental problems in science and engineering whose solutions could not be contemplated in the absence of such power. For some of these problems, the *Grand Challenges*, we stand on the threshold of computational feasibility. We are close to being able to perform computer simulation of certain experiments that are undesirable to perform physically (e.g., those involving the use of animals or toxic chemicals). We are beginning to collect, and are almost able to store and provide access to, the enormous quantities of data, such as detailed world climate data, necessary for analysis and eventual understanding of the greenhouse effect, ozone depletion, and other large-scale phenomena of critical importance to energy policy.

In section 2 we provided some indication of the changes in individual worklife being wrought by information technology. A further change is the increased skill levels necessary to perform the work that follows from the introduction of sophisticated technology into the workplace. We shall consider some of the implications of this at the end of this section.

3.1.2 Demand for High Performance Computing

We must move from MGM territory into TTG territory...

The demand for high-performance computing in DOE and other government agencies is well documented [1,2,3,4]. High-performance DOE applications include climate modeling, fluid turbulence, pollution distribution, mapping the human genome, ocean circulation modeling, quantum chromodynamics, semiconductor modeling, superconductor modeling, and combustion systems modeling.

These applications need high performance in all aspects of the computing universe. Today's generally-available

high-performance computers operate in what we might call MGM territory (they provide megaflops—millions of floating-point operations per second, gigabytes—billions of bytes of dynamic storage, and megabits—millions of bits per second for data communications); we need to move into TTG territory (teraflops, terabytes, and gigabits, where tera- denotes trillions).

...and develop the software to manage it.

DOE also requires high-performance software, both to manage the complexities of the new hardware architectures, and to provide the kind of assistance the scientists will need. As a general rule, the most advanced software systems that are are now available are those on personal computers. Software of this caliber must be integrated into the grand challenge computing environment.

3.1.3 Industry Status and Trends

Economic competition continues to drive information and computing technology toward increased performance and lower costs.

In our fascination with technological innovation, we tend to forget that the most potent driver of this technology is not the challenge of the problems but the economic incentive. It is no longer the case, for example, that the most "bang for the buck" comes with the largest machines, as it did as recently as ten years ago. Today, microprocessor chips have this distinction not because they are faster, but because they are so much cheaper. (This could change if the market for floating point operations changes, since the most important single factor in determining cost is market size and the savings brought about by mass production.)

Industry has recognized the potential demand, however, and there are encouraging signs that the price penalty for high performance will decrease substantially in the next few years.

Achieving computational performance through parallelism.

One approach to providing supercomputer performance capacity at lower cost is to use many low cost floating point units in parallel. Such systems are now becoming commercially possible. Teraflops parallel machines have been proposed and implementation has begun. Parallel computers based on medium speed floating point units now deliver performance roughly comparable to the top of the line supercomputers. The cost of flops on such systems

systems is significantly lower than the cost on traditional supercomputers. However, it is not easy to break problems up into pieces that can be performed in parallel. Programming languages that allow compilers to fully exploit parallelism have not yet evolved, and there is a growing belief that such languages will have to be problem-area specific.

In the meantime, performance measurement tools are starting to emerge that allow programmers to identify the bottlenecks in parallel operation. Often it is possible to remove a bottleneck by changing the code slightly, but this is a costly and time consuming process, requiring a person with a deep understanding of both the problem and the available parallel processing methods.

Data storage. The per-bit cost of data storage currently depends upon the the speed with which data can be stored and retrieved. In general the greater the speed the greater the cost of the storage device. We consider this area in more detail in a succeeding subsection of this Assessment.

Data communications. The cost of data communications technology depends upon speed and distance between the sender and the receiver. As a rule, one pays more for greater speed and greater distances.

The top end of data communications performance is provided by computer backplanes. The backplane of a modern supercomputer is capable of delivering data at gigabyte/second rates. Scientific workstations operate in the range of megabyte/second. Older style personal computers are in the range of kilobyte/second. The maximum distance for backplane technology is a few feet. Thus, distance is not a significant cost factor for backplanes.

Local area networks represent the next performance niche. Here performance typically ranges from hundreds of megabytes per second to a megabyte per second or so. The maximum distance for local area networks is a few kilometers. Distance costs for local area networks tend to be dominated by costs of laying cable, not by the cost of

the cable itself. For medium distances, this technology provides excellent performance for very low cost.

Wide area networks have traditionally represented the lowest data communications performance. However, this is changing with the growing national network of fiber optic cables. The high end performance range of wide area networks now overlaps the mid range of local area networks. For example, it is expected that NREN will use 45 megabit/second long distance lines. The cost of wide area networks is usually dominated by the cost of leased communication lines.

The most sophisticated software tools are not easily accessible to Grand Challenge problems.

The personal computer market receives the lion's share of commercial software development dollars. Sales of this software have been estimated at eighty billion dollars per year. As a result, the most sophisticated software available in the world currently runs on personal computers. Examples include spreadsheets, symbolic mathematics, data management tools, project management tools, inventory management, and publications tools. All of these are useful and are heavily used in solving grand challenges. But at present, integration of grand challenge software and data with these mainstream commercial products remains difficult.

3.1.4 Projections

3.1.4.1 Short Term Projections

High performance computing: the supercomputing user community may shrink, but demand will remain vigorous.

As lower-cost systems achieve higher and higher performance, they will inevitably attract Grand Challenge applications away from higher cost systems. This will not happen rapidly because of the large effort and high skill levels required to move from one system to another. The number of users of the nation's supercomputer centers such as DOE's National Energy Research Supercomputer Center (NERSC) will therefore gradually decline. The remaining users will nevertheless demand more cycles than ever before. To meet the needs of some of these users, special purpose high performance systems will continue to be developed as research vehicles. Several such projects were described at the 1989 Symposium on Lattice Field Theory [5,6,7,8].

Multiprocessor performance will exceed traditional mainframes for many grand challenge problems in the next two to five years. Multiprocessors will move in alongside traditional supercomputers in the supercomputer centers and will begin to take over significant portions of the workload.

Effort and skill levels needed to optimize programs to run efficiently on these systems will remain high. However, the job mix as a whole should run fairly efficiently from the outset. The main incentive for optimizing individual codes will be to reduce turnaround time for individual problems.

Data storage: two orders of magnitude increase in demand. No short-term solution in sight.

The demand for mass storage by the Grand Challenges will grow significantly. high-energy physics, climate, waste management, and fusion will all generate increasing demand for mass storage. For example, colliding beam experiments at Fermi National Accelerator Laboratory currently produce data at about 15 gigabytes per day. Corresponding rates for the Superconducting Super Collider are projected to be 1-10 terabytes per day. There is no known mass storage technology that would allow this quantity of data to be stored online at acceptable cost [9]. NASA's Earth Observing System (EOS) project has similar requirements [10].

NERSC and the other national supercomputer centers will continue to be major users of the high end of this technology.

Data communications: Fiber and Futurebus+.

The demand for both local and wide area data communications will increasingly be met by FDDI (Fiber Distributed Data Interface) technology and by twisted pair Ethernet technology. FDDI currently operates in the 100 megabit per second range, while Ethernet operates in the 10 megabit per second range. Products based on the SCI (Scalable Coherent Interface) standard should also begin to appear in this time frame.

In the computer backplane performance range, VME will continue to dominate scientific applications. However, products based on the Futurebus+ standard will start to appear. These will be used to build the most demanding

data acquisition systems such as those required by the Superconducting Super Collider and the Relativistic Heavy Ion Collider.

Sophisticated software tools: limited by lack of industry interest.

Office tools such as spreadsheets, CAD tools, project management tools, and data management tools will increasingly be integrated into Grand Challenge computing problems. For example the CDF detector at Fermilab already uses Macintosh computers in the control system as well as for analysis and display of physics results. However, integration effort will not be well funded by industry and progress will be correspondingly slow. The Scientific Computing Staff of DOE's Office of Energy Research has funded research into supercomputer access techniques [7] in an effort to accelerate this process.

Performance analysis tools will become easier to use and more widely available as providers of high performance systems compete to make their products more cost effective. Fortran compilers will be able to exploit parallelism to the limits imposed by their algorithms.

3.1.4.2 Long Term Projections

At the high end, fewer users will need more total power; at the low end, explosive growth in demand and using population; increasing competition for technical personnel.

DOE's need for high-performance, high-cost information and computing technology will continue with an increasing demand by a diminishing group of users. By contrast, DOE's need for medium and low cost information and computing technology will grow with a sharply increasing demand by an expanding group of users. The total DOE investment in information and computing technology will increase significantly.

The pressure on industry to provide the highest performance technology at the lowest possible cost is enormous. It has created a significant market for highly skilled technical and scientific people. These are the same people needed by government to attack the scientific Grand Challenges. Industry is thus competing with government for this workforce. The inevitable result is that government will gradually lose its best technical and scientific staff to industry.

This combination of fewer DOE individual users of high performance computing technology with unprecedented needs for performance, an increased dependence by grand challenge science on commercial software products, and a migration of the skilled scientific and technical workforce from DOE into industry will have a significant impact on the future of DOE scientific and weapons programs. Some adjustment of the relationship between DOE and industry may be desirable: e.g., greater involvement of industry in designing and building scientific apparatus. The DOE component of the Federal High Performance Computing and Communications Program [2] outlines strategies for achieving this goal.

References

[1] The Committee on Physical, Mathematical, and Engineering Sciences, "Grand Challenges: High Performance Computing and Communications", *The FY 1992 U.S. Research and Development Program,* A report to supplement the President's fiscal year 1992 budget, Computer and Information Science and Engineering, 1800 G Street, N.W., Washington, DC 20550.

[2] US Department of Energy Office of Energy Research Scientific Computing Staff, "The DOE Program Component of the Federal High Performance Computing and Communications Program", Washington, D.C. 20585, June 1991.

[3] U.S. Congress, Office of Technology Assessment, "High Performance Computing and Networking for Science - Background Paper", *OTA-BP-CIT-59* (Washington, DC: U.S. Government Printing Office, September 1989).

[4] U.S. Department of Energy Office of Administration and Human Resource Management, "FY 1992 - FY 1996 Information Resource Management Long-Range Plan", Associate Director for Administration, Information, and Facilities Management, Office of IRM Policy, Plans and Oversight, Washington, DC 20585.

[5] Christ, N.H., "Status of the Columbia 256-Node Machine", *Lattice 89: Proceedings of the 1989 Symposium*

on Lattice Field Theory. Edited by Nicola Cabibbo and others. Amsterdam, the Netherlands: Elsevier Science Publisher, 1990, pp. 267-271.

[6] Iwasaki, Y. and others, "Status of QCDPAX", *Lattice 89: Proceedings of the 1989 Symposium on Lattice Field Theory*. Edited by Nicola Cabibbo and others. Amsterdam, the Netherlands: Elsevier Science Publisher, 1990, pp. 259-262.

[7] Fischler, M. and others, "Designing Machines for Lattice Physics and Algorithm Investigation", *Lattice 89: Proceedings of the 1989 Symposium on Lattice Field Theory*. Edited by Nicola Cabibbo and others. Amsterdam, the Netherlands: Elsevier Science Publisher, 1990, pp. 263-266.

[8] Weingarten, D., "The Status of GF11", *Lattice 89: Proceedings of the 1989 Symposium on Lattice Field Theory*. Edited by Nicola Cabibbo and others. Amsterdam, the Netherlands: Elsevier Science Publisher, 1990, pp. 272-275.

[9] Baden, A. and others, "Object Oriented Data Base Requirements for High Energy Physics", LBL internal report, August 1991.

[10] Dozier, J., Looking Ahead to EOS: The Earth Observing System", *Computers in Physics,* May/June 1990.

3.2. Scientific Data Management

3.2.1. Introduction

Scientific data are typically complex and managed by special-purpose, ad hoc *programs instead of by data management systems.*

The management of scientific data imposes requirements that differ in several respects from the management of comercial data. Generally speaking, scientific databases are characterized by the existence of complex data structures and a propensity for long transactions; neither of these attributes is well served by commercial database management systems. As a result, scientific databases

(SDB's) typically are stored as files that are then managed and processed by special-purpose programs written specifically for each application.

Technology encourages the creation of larger databases, but there exists much data that is inaccessible because we don't have adequate management tools.

Technological advances are encouraging the creation of larger databases both directly—by the utilization of more sophisticated devices that generate more data—and indirectly—by making cheaper computer power available (thus allowing the data to be processed). A single scientific experiment can generate hundreds of megabytes of data within days. Even so, many scientific simulations are not carried out to the desired granularity level because it is impractical to process and manage the large amounts of data that would be generated. In addition, there are large amounts of data that were not collected originally for experimental or statistical purposes, but have tremendous potential when used for statistical purposes. For example, routine patient records in hospitals can be used for statistical "cause and effect" studies. For the most part, such sources of routine collections of data are left unused because adequate data management facilities do not exist.

3.2.2. Limitations of Commercial Data Management Systems

The benefits of data management systems.

Data management technology has been used successfully in many applications (mostly business oriented) because it offers the following advantages:

a) data models that provide an abstraction for representing the structure and semantics of the data;
b) high-level query languages defined over the data models for accessing and manipulating the data;
c) support for concurrent access to the data by multiple users (concurrency control);
d) mechanisms for expressing and validating the integrity of the data;
e) back up and recovery of the database as protection from system failures; and
f) support for efficient physical organization in terms of storage on and access from secondary and tertiary storage.

In general these advantages could be useful for SDB's as well as for business applications. However, in each of the areas noted above there are specific additional or different requirements for SDB's.

The Data Model

The relational model is unsuited to SDB's.

The data model most used in modern commercial data management systems is the relational model. It consists of multiple tables (called "relations") that can be linked explicitly in the query language by associating columns (a "join" operation). The model is set oriented, that is, the order of instances (rows of a table) is not enforced. This simple table structure is inadequate for SDB's. For example, ordered structures, such as DNA sequences or temporal sequences, cannot be directly represented as relations. Other data types, such as vectors, matrices, etc., also cannot be expressed by relations in a straightforward and convenient way. SDB's also often include graphs, images, and complex tables (e.g., isotope tables) that are not well suited to the relational model.

Another important requirement is the ability to represent complex objects consisting of structured arrays of individual data elements, such as images or contours in 2- or 3-dimensional space, or sequences of such images or contours. Such complex object structures are prevalent in SDB's, but their representation in terms of relational tables is quite awkward, complex, and unnatural.

The Query Language

SDB's need a more powerful language than SQL.

SQL (Structured Query Language) has become the *de-facto* standard high-level query language for the relational model. The basic functionality is provided by constructs for specifying predicates to select rows from relational tables, specifying how multiple tables can be associated (joined) in a single query, and which columns should be included in the output. Additional features of the language include aggregate functions (e.g., count, sum) over groups of values, set operators (union, difference), and sorting of the output. SQL is a fairly powerful language for expressing data manipulation of table structures, although there is continuing criticism of its

semantic clarity and power. In particular, SQL does not support the specialized, discipline-specific requirements of SDB applications. For example, many scientific data models make use of a sequence construct to represent seismic events in time, text sequences, DNA sequences, etc. In general, each of these domains requires different operators. For seismic sequences we may be interested to find if an event occurred at a particular range of time, for text sequences we may be interested in finding words with certain proximity to each other, and for DNA sequences we may want to find overlapping regions between them. Operations of this type are not supported by current versions of SQL.

Concurrent access

Concurrency is less of a problem for SDB's than for commercial databases...

Support for concurrent access to data for business applications revolves around the concept of a transaction. The paradigm used is that multiple transactions may be applied to the same data item(s) and thus interfere with each other when at least one transaction updates the data. For business applications, such as banking, accounting, or reservation systems, such interference may be disastrous.

By contrast, concurrent access to the data is not as important an issue for SDB's. First, much of the data is historic, rather than dynamic, and so, once taken, is not updated (e.g., data generated by scientific devices, such as those used in weather monitoring). Second, if data is updated, it is often done by a single person, as is the case when an analyst removes or corrects outliers (data that is outside expected bounds). In such a case, the corrections do not have to be immediately visible, and thus can be made on a locked version of the database. Third, when multiple users need to update the data simultaneously, they often need to access different parts of the database, as would, for example, multiple engineers working on different sections of a joint design. The concept of a "long transaction", however, is often more appropriate for SDB's, where a scientist may embark on a long analysis

...but long transactions are more of a problem.

(which results in updates). It is not always possible to break such long transactions into a series of short transactions.

Integrity constraints

Integrity constraints for SDB's differ from those required for commercial databases.

Maintaining the integrity of data can be quite complex. Most commercial systems support a simple form of data-type integrity, such as checking that values not exceed given boundaries, that values occur only in a certain format, or that values be wholly numeric or wholly alphabetic. SDB's require generalized versions of these features, such as that admissible data values are limited to those from a given list of categorical values.

Recovery

SDB's need recovery mechanisms that support recovery in the midst of long transactions.

Mechanisms for back up and recovery are always needed. In conventional applications where transactions are short, it is quite acceptable to back up to the last successful transaction for the purpose of recovery. For SDB's, where long transactions are more typical, backing up to the last transaction is not acceptable, because too much work may be lost. New mechanisms need to be developed for the support of long transactions.

Physical database organization

SDB's require organizations that support clustering by physical or temporal locality, or by similarity of components, and that provide efficient support for sparse data.

Physical data structures and access methods are the key to efficient support of queries. Relational database implementations typically organize rows of relations as records in files, and provide additional index mechanisms (e.g., B-trees, hash tables) over the various columns of the relations. SDB applications need additional types of data organization and access algorithms. Applications involving models might benefit from an organization based on spatial or temporal locality; others, or search or compression strategies for sparse multidimensional data; still others, from column-wise (rather than row-wise) organization.

It is worth noting also that complex objects have a natural clustering of their components. For applications that

need to access entire complex objects, object-wise clustering would be best.

As can be seen from the above examples, data management systems for SDB's need to have a rich set of physical organization options to provide efficient performance. A multiplicity of choices makes the problem of query optimization (i.e. the algorithm for the most efficient way to execute a query) so much harder. The management of these options and the way they can interact are a great challenge to designers of such systems.

3.2.3. Current Approaches and Their Relationship to SDB's

New approaches to the management of complex databases may provide better support for SDB's.

There are several promising new approaches that could have an effect on the future management of SDB's. These were not specifically developed for SDB's, but for complex database applications in general. However, the capabilities they are designed to provide can go a long way toward supporting SDB applications. In general, these approaches strive to provide the following capabilities:

1) support for complex objects, and operations over them,
2) support for user defined structures and functions, and
3) the capability to incorporate new physical data structures and access methods.

It follows from the discussion and examples in the previous section, that such capabilities are indeed necessary for the efficient support of SDB's. The approaches we discuss below are fundamentally different from each other, thus emphasizing different aspects of the above capabilities.

Object-Oriented Database Systems

Simplification by aggregation: defining complex entities as objects.

The two main features of object-oriented development are *encapsulation* and *inheritance*. *Encapsulation* consists of the ability to define data structures and operators over them (also called "methods"), and make them available to

users through interfaces. The objective is to hide the details of the implementation of the data structures and operations and make only their interfaces visible. Accordingly, object structures in SDB's, such as a seismic event, could be encapsulated, together with specialized operators, such as "find other events that correlate with this event".

The usefulness of this concept to SDB's is in the power to code new complex structures and operations, and make them callable by name. For example, a DNA sequence could be coded as a linked list structure, and the operation "overlap" could be coded to return the sub-sequences that overlap between two given DNA sequences. Both the DNA structure and the overlap operation will then be recognized by the system, and could be invoked by name.

Inheritance supports the construct of classes and subclasses (e.g., a student is a subclass of the class person), and provides the inheritance of its properties (e.g., the age of the person is inherited by the student). In addition, object-oriented inheritance includes the inheritance of the operators (methods) associated with the parent object. This feature provides the capability to share code and to make explicit the inheritance association between objects.

In current approaches to object-oriented database management systems (OODBMS's) support for encapsulation and inheritance is provided by the underlying programming language, such as Object-Pascal or C++, rather than by the OODBMS itself. Unfortunately, there are as yet no mechanisms to make the new objects and structures part of some high level query language (they are only callable from a program), and no mechanisms for query optimization.

User-Defined Structures and Functions

Adding object-oriented concepts to an RDB.

There have been various attempts to extend the relational data structures and relational query languages (such as permitting relations made of relations), and to design systems that can include user-defined data structures and operations. The typical approach in such extensions is to

define a complex relational model and a query language for it. The extensions include inheritance structures, new data types (e.g., vectors, matrices, images, etc., and even procedures), and a capability to construct complex relations from other relations. Further, the systems are designed to be extendable, in that the user can define new structures and operations in terms of data structures and operators provided by the system. The user-defined operations are callable by name from the query language.

It is too soon to predict whether such attempts will succeed. While they show promise, there is some indication that query optimization in such systems may not be generally possible for the database user, but may require the continued attention of a specialist.

New Physical Data Structures and Access Methods

Application-specific extensions to DBMS kernels.

Extensible database systems are based on the premise that a single system cannot be designed to answer the complex needs of various applications. Thus, they are designed to customize specific data management system for each application. The main idea is one of using building blocks, that is, reusable software that can be selected for a particular application. Additional blocks can be provided by users, and the selected blocks are then assembled (compiled) to produce a special purpose system customized for the application. The key to the success of this approach is the design of interfaces that permit the interchangeability of modules and the integration of new modules.

Obviously, this approach should be quite attractive for scientific database applications. However, the designer of such a customized system has to have sufficient expertise in the implementation of DBMS's. Thus, this approach may be considered more appropriate to a software house that can customize a DBMS for the needs of an application.

Extensible database systems have to address explicitly the issue of incorporating new operators, new query languages, and new physical data structures and access methods. While these can be introduced as "plug

compatible" modules, there is still the problem of having to modify the query optimizer accordingly. One of the proposed solutions is to use rule-based optimizers to specify the optimization algorithm.

3.2.4. Implications and Prognosis

The situation will both improve and stay the same.

Experience from the commercial world indicates that progress in the development of data management technology is slower than most predictions would indicate. Nevertheless, we can see real possibilities for the development of data management systems suitable for the management of SDB's. Data-intensive programs such as mapping the human genome are spurring the necessary research efforts, and, as indicated above, some useful extensions to conventional technology are beginning to emerge. As has been the case in other areas of ITR, the hardware technology that enabled the creation of very large databases has preceded the software technology for their management. We expect this trend to continue with the introduction of multilevel, multistream mass storage systems. Thus we can expect the next several years to be characterized by a duality, in which, on the one hand, significant progress is made in the introduction of true data management technology for the management of conventional SDB's resident on single-level devices, but, on the other, a continuing *ad hoc* approach for certain SDB's involving unusual organizational requirements or resident on advanced storage technology.

3.3. Analysis Tools

3.3.1. Statistical Analysis

A quick survey of existing tools,...

Advances in statistical analysis over the last decade have been largely in areas requiring considerable computing. Some of these are

- The hard cases of standard models, where estimators cannot be written down in closed form, such as unbalanced mixed linear models, data that are censored or truncated, and methods that are robust against failures of assumptions. For easy cases, estimates have closed forms; for these harder cases

estimates must be obtained by numerical minimization.

- Bayesian statistics, where only the most trivial cases can be given in closed forms. These methods allow one to use stronger information (the relative desirability of various outcomes and their relative likelihood) to draw stronger conclusions about unknowns (actual probability distributions for estimated quantities).
- Semi-parametric and non-parametric methods, such as the fitting of splines to observed data. These methods give fitted curves in situations where no simple model is close to correct.
- Resampling methods such as bootstrapping and cross-validation. These eliminate assumptions about distributional form by repeatedly sampling subsets of the data, at the cost of doing a specific analysis many times (to get a bootstrap estimate of a variance requires as many analyses as there are points in the original analysis; to get the uncertainty of that estimate squares the number of analyses).
- Exploratory methods with considerable interactive and graphical components. These are the discrete data analog to visualization techniques for continuous data. There is the added demand of working interactively, rather than off-line as complex visualization techniques often do.

...how they impinge upon the work of the analyst,...

The order of the above examples was chosen to range from those that use computing only by necessity, and that could exist in principle in a batch-oriented world, down through those for which all aspects of the modern computational environment are necessary. These modern techniques of statistical analysis have been made possible largely by improvements in the infrastructure of computing.

The ability to devise powerful parser-driven languages has made it possible for researchers to experiment with different methods without being constantly mired in coding in a low-level programming language.

Most major data analysis packages have some limited unilateral interoperability (i.e., they can run Fortran or C code or execute shell scripts that can run other programs), so that users are not trapped by limitations of their interactive languages or slowed permanently to the speed of an interpreter. The explicit networking now available has made it possible to use a large, fast machine or very large archival stores from distant sites, which has made practical the economies of scale required for computationally intensive operations and applications. Interactive graphics with brushing and point identification have played an important role in the development of exploratory techniques. Many of the ideas for modern statistical techniques have been around for years, but these infrastructural improvements were on the critical path to their realization.

While modern techniques are widely accepted among statistical researchers, they have yet to gain much ground among users. Most easy-to-use statistical packages do not provide access to modern methods, but rather are simple ways to get at the statistical technology of the late sixties. Most scientific applications for data analysis also omit any but the most rudimentary statistics. As an example, many programs provide the calculations for specific cases of nonlinear least squares; few provide parameter uncertainty estimates, and those few use older asymptotic methods for those estimates. While this gap between the state of the art and common practice is partly just the time lag that would normally be expected, some difficulty in introducing modern methods to general practitioners is due to the incompleteness of the infrastructural changes discussed above.

...and how the situation could be improved.

For scientists to begin having access to modern statistical methods, more computational infrastructure is required. Here are some examples of the kinds of limitations present, and how they can be lifted:

- While powerful parser-driven languages are available, the path from interpreted language to rapid compiled code is still difficult, limiting code in high-level languages to a slower speed than Fortran

or C code. This puts an artificially low limit on the amount of computation deemed feasible.

- The development of interactive graphics is straightforward, while the development of graphical interfaces is less so. The former can be done by any practitioner familiar with a modern statistical package; the latter, even when done with the latest tools, is basically a batch process that requires extensive hand coding in C or C++. This is related both to the point above and the one below.
- Interoperability is limited and unilateral. A statistics package can use C code; rarely is the reverse true. Database management systems can usually be called from C code, but DBMS's rarely can access any functionality beyond their query languages, which are crippled from the point of view of scientific computation. The availability of complete bilateral operability would enable the basic platform to be the one most appropriate, rather than forcing work in C because the database can only be effectively accessed there, or causing the production of extensive and awkward macros in an interactive package because the graphics provided with the package cannot be pulled into a better, custom-written interface.
- Networking is opaque and forced. If one wants to do computing that is essentially beyond the capabilities of one's own machine, the only option is to move code and data to a larger or faster machine. In resampling methods, an order of magnitude of computational power or storage space may be required only temporarily. In interactive exploratory methods, a nearest-neighbor algorithm searching over many thousands of points may be essential on rare occasions. Translucent networking that permits higher-powered computing to be accessed on demand or transparent networking which recognizes the need for such power and obtains it automatically would drop a prominent barrier to modern, computationally intensive methods.

Organizations like DOE can make a substantial difference, both because they have large projects whose computational requirements vary widely even within a project and because DOE is a big user of computers and services, and so can require improvements of vendors. The end result, if properly encouraged, could be to put the toolbox of a current research statistician at the disposal of most general scientists within the next few years.

As we have seen in section 2.7, the needed improvements in infrastructure are on the way. What is not yet clear is how quickly the providers of statistical analysis software will take advantage of them.

3.3.2. Graphics Workstations

On tap for graphics workstations: more speed, wider choice of input devices, better color, and further advances in interface software.

While the physical components of a standard graphics workstation have not changed much in the last five years, its capability has increased dramatically. The next three to five years should see continued increases in speed (on the order of 20-40% every six months), and a number of innovations in input options. Currently the only choices for input are the mouse (or trackball) and the keyboard. Video input touch screens will become available within three years, followed shortly by voice recognition input. This will make workstations more accessible and easier to use for all of us, especially for those with physical disabilities. Three-dimensional input devices are also now becoming available, and should soon have a greater impact in the graphics market.

Color has always been important to computer graphics and the current hardcopy devices are making it more practical and realistic. As these devices get cheaper, the use will of color will expand. The advent of high density television (HDTV) will allow the high-quality color output we see on our workstation to be transferred as video data. This will make the creation of scientific videos easier and more practical, and make the translation to the national television broadcast standard (NTSC), with its low color and spatial resolution, unnecessary.

Despite the impressive hardware changes, the most important advances in graphics workstation technology

have been in software, particularly the user interface. The X-Window system allows users to distribute their graphics applications across multiple machines. The next major advance will be in the integration of cooperating applications. Still too many applications act differently, expect different types of input, and do not communicate. Users are demanding more standards through their pocketbooks and user groups.

Graphics workstations will have their own multimedia revolution. Video conferencing will be available at a user's desk in the next three years and will radically affect how scientists work. Small, *ad hoc* get-togethers with colleagues will be possible, in which each participant will have a window on the others' workstations that will show either the person (live) or a view of the room or desk. Other windows will be available for sharing document editing, graphics output, etc. This will make electronic collaboration easier and more productive. It is too soon to tell whether this small-scale video-conferencing capability will replace room-size video-conferencing facilities, or enhance their appeal. Past experience suggests the latter.

In the shorter term, we can expect to see multimedia extensions (voice, pictures, and video) to e-mail and other file-transfer services.

3.3.3. Visualization

Visualization remains hampered by a lack of standards. Nevertheless, we will see visualization employed in real-time control systems, and it, too, will develop multimedia extensions.

Imaging and visualization have begun to come into their own in the last three years, and form a growing market. Effective visualization depends on many fields, including traditional computer graphics, imaging, database management, high-speed networks, innovative input and output, computational and parallel processing algorithms, and user interface design. The lack of standards in these areas, both individually and as aids to interoperability, have increased their difficulty for users, and consequently slowed their dispersion into the marketplace.

In the next three years we expect to see the emergence of environments that are more integrated, robust, and transportable. User groups have been formed to share visualization modules, but the integration between

different packages is still non-existent. In addition, more and better tools are needed, as well as faster and better rendering.

In the next five years we should see real-time visualization systems used as front ends for controlling experiments. Users will be able to analyze the data from an experiment and modify the experiment based on this real-time analysis.

Use of sound in visualization is still relatively new and not used widely. Our ears are finely tuned to small changes, however, and in the next five years we should see sound as well as sight being integrated into visualization packages. This area should benefit from the research and development now underway into human interaction with complex multimedia data sets (virtual reality).

3.4. Integration Tools and Interoperability

The distribution of large applications over collections of heterogeneous platforms is desirable.

Workstations and computer systems have been interconnected by various networking schemes for some time. Although these architectures are "distributed" in the sense that different portions of program execution take place on different processors, the division of labor between systems is often a very coarse one—typically between disk storage operations and non-I/O processing. The task of truly distributing portions of an application over several (or many) computing platforms remains a very difficult and laborious one. However, the creation of a large application program that spans multiple, heterogeneous computer systems and appears to function as a single, well-integrated object remains a desired goal.

One advantage of such a procedure is the ability to match each segment of the problem to an appropriate resource.

One motivation for finer-grained distributed applications is the desire to match different aspects of a program's behavior to appropriate computing resources. For example, it is desirable to have the compute-intensive portion of a combustion modeling program execute on a supercomputer while the display and manipulation of graphical

output take place on a high-speed workstation. In some current applications, a degree of functional decomposition has been achieved, but the labor required to make such a software package appear as a single, well-integrated application remains substantial.

Another is that some resources are available only on certain platforms.

A second motivation comes from the need for greater flexibility in assembling applications from existing functional program parts—both commercial packages and application-specific codes. At present, it is difficult to successfully blend portions of existing commercial packages with pieces of user-written code to produce a single, well-integrated application. The lack of industry-wide, consistent program interfaces, program calling disciplines, and data interchange formats make the interconnection of different software packages a laborious and, therefore, expensive process.

Object-oriented techniques will facilitate the process.

To realize both of the goals outlined above, the software industry is proposing the application of object-oriented program techniques to distribute programs across multiple platforms. These techniques will give programmers a new level of flexibility in creating customized applications. One such distributed application could be a next-generation high-energy physics detector-modeling system. In such a system, objects from a commercial mechanical CAD system would be linked both to one or more physics simulation packages and to objects of a commercial data visualization package. The effects of mechanical design changes and material selection could then be quickly seen in terms of their physics consequences and detector performance. Furthermore, a single object-oriented database system would provide archival storage and revision management services for all the system's components. Although some of these capabilities are available now, they lack a robust, distributed programming environment; without such an environment, programs cannot achieve the level of seamless integration needed to make them function as a single application.

There are several industry-wide efforts addressing this problem. Those that hold the most promise are based on the industrial consortium model, where supporting

members exchange designs and technology during the "precompetitive" phase of development. In this regard, two organizations stand out: the Object Management Group (OMG) and the Open Software Foundation (OSF). Each of these groups is assembling the basic software technology to allow user applications to be distributed across a wide variety of hardware and software platforms.

When these techniques become commercially viable (within the next two years), the research community will have much to gain from their adoption. DOE programs should participate in these organizations and, at the earliest opportunity, sponsor implementation of proof-of-principle distributed, object-oriented demonstration projects.

3.5. Sample Hardware Architectural Developments

3.5.1. The Storage Hierarchy and Storage Technology

The storage pyramid, 1980: minicomputer with cache memory.

The storage hierarchy is traditionally modeled as a pyramid, with a small amount of expensive, fast storage at the pinnacle and larger capacity, lower cost, and lower performance storage as we move toward the base. In general, there are order-of-magnitude differences in capacity, access time, and cost among the layers of the hierarchy. For example, main memory is measured in megabytes, costing approximately $50/megabyte, and can be accessed in small numbers of microseconds. Secondary storage, usually implemented by magnetic disk, is measured in gigabytes, costs less than $5/megabyte, and is accessed in tens of milliseconds. The operating system can create the illusion of a large fast memory by judiciously staging data among the levels. However, the organization of the storage hierarchy must adapt as magnetic and optical recording methods continue to improve and as new storage devices become available.

Figure 3 depicts the storage hierarchy of a typical minicomputer of 1980. (It should be noted that large mainframe and supercomputer storage hierarchies were more complex than what is depicted here.) A small file cache (or buffer), allocated by the operating system from

the machine's semiconductor memory, provides the fastest but most expensive access. The job of the cache is to hold data likely to be accessed in the near future, because it is stored near data recently accessed (spatial locality) or because it has recently been accessed itself (temporal locality). *Prefetching* is a strategy that accesses larger chunks of file data than requested by an application, in the hope that it will soon access spatially local data.

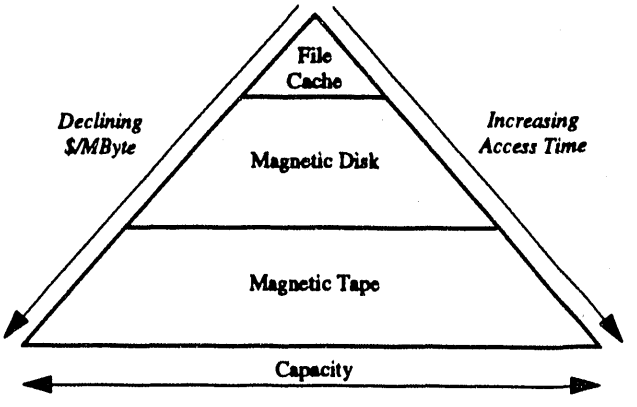

Figure 3
Typical Storage Hierarchy, Circa 1980

Microsecond access is provided by the file cache, a small number of bytes stored in semiconductor memory. Medium capacity, denominated in several hundred megabytes with tens of millisecond access, is provided by disk. Tape provides unlimited capacity, but access requires times on the order of tens of seconds to minutes.

Either a buffer or a cache can be used to decouple application accesses in small units from the larger units needed to utilize secondary storage devices efficiently. It is not efficient to expend the millisecond latency cost to access secondary storage for a small number of bytes. Accesses in the range of 512 to 8192 bytes are more appropriate. The primary distinction between application memory and a cache is the latter's ability to keep certain data resident, such as frequently accessed file

directories, to avoid slow accesses to the lower levels of the hierarchy.

Secondary storage is provided by magnetic disk. Data are recorded on concentric tracks on stacked platters, which have been coated with magnetic materials. The same track position across the platters is called a cylinder. A mechanical actuator positions the read-write heads to the desired recording track, while a motor rotates the platters containing the data under the heads.

Tertiary storage, provided primarily for archive/back-up, is provided by magnetic tape. A spool of magnetic tape is drawn across the read-write mechanism in a sequential fashion. A good rule of thumb for a unit of tertiary storage media, such as a tape spool, is that it should have as much capacity as the secondary storage devices it is meant to back up. As disk devices continue to increase in capacity, tertiary storage media are driven to keep pace.

In 1980, a typical minicomputer would have had one to two megabytes of semiconductor memory, of which only a few thousand bytes might have been allocated for input/output buffers or file system caches. The secondary storage level might have included a few hundred megabytes of magnetic disk. The tape storage level, then as now, was limited only by the amount of shelf space in the machine room.

The storage pyramid, 1990: multiple caches plus LAN.

Figure 4 shows the storage hierarchy distributed across a workstation/ server environment of today. Most of the semiconductor memory in the server can be dedicated to the cache function because a server does not host conventional user applications. The file system "meta-data," that is, the data structures describing how logical files are mapped onto physical disk blocks, can be held in fast semiconductor memory. This represents much of the active portion of the file system. Thus, disk latency can be avoided while servicing user requests.

The critical challenge for workstation/server environments is the added latency of network communications. Network latency times are comparable to those of magnetic disk, and are measured in small tens of

milliseconds. The figure shows one possible solution, which places small high performance disks in the workstation, with larger, potentially slower disks at the server.

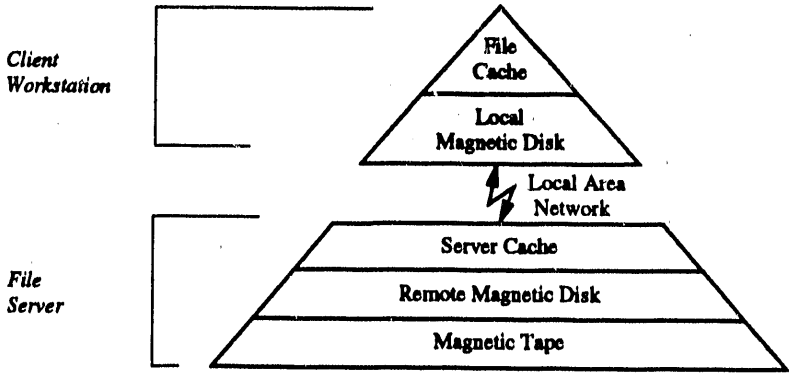

Figure 4
Typical Storage Hierarchy, Circa 1990

The file cache has become substantially larger, and may be partially duplicated at the client in addition to the server. Secondary storage is split between local and remote disk. Tape continues to provide the third level of storage.

If most accesses can be serviced by the local disks, the network latencies can be avoided altogether, improving client performance and responsiveness. However, there are several choices for how to partition the file system between the clients and the servers. Each of these partitionings represents a different tradeoff between system cost, the number of clients per server, and the ease of managing the clients' files.

Swapful, dataless, diskfull, and diskless clients.

A *swapful* client allocates the virtual memory swap space and temporary files to its local disk. The operating system's files and user files remain on the server. This

reduces some of the network traffic to the server, leaving the issues of system management relatively uncomplicated. For example, in this configuration, the local disk does not need to be backed up. However, execution of an operating system command still requires an access to the remote server.

A *dataless* client adds the operating system's files to the client's local disk. This further reduces the client's demand on the server, thus making it possible for a single server and network to support more clients. While it is still not necessary to back up the local disk, the system is more difficult to administer. For example, system updates must now be distributed to all of the workstations.

A *diskfull* client places all but some frequently shared files on the client. This yields the lowest demands on the server, but represents the biggest problems for system management. Now the personal files on the local disk need to be backed up, leading to significant network traffic during backup operations.

An alternative approach leverages the lower cost semiconductor memory to make feasible large file caches (4 to 8 megabyte) in the client workstation. These "client" caches provide an effective way to circumvent network latencies, if the network protocols allow file writes to be decoupled from communications with the server (see the discussion of NFS protocols in the next section). The approach, called *diskless* clients, has been used with great success in the Sprite Network Operating System [1], where an ability to support 5-10 times as many clients per server as more conventional client/server organizations is reported.

The storage pyramid, 1995: disk arrays, near-line storage, LAN, and WAN.

Figure 5 depicts one possible scenario for the storage hierarchy of 1995. Three major technical innovations shape the organization: disk arrays, near-line storage subsystems based on optical disk or automated tape libraries, and network distribution. We concentrate on disk arrays and near-line storage system technology in the remainder of this subsection.

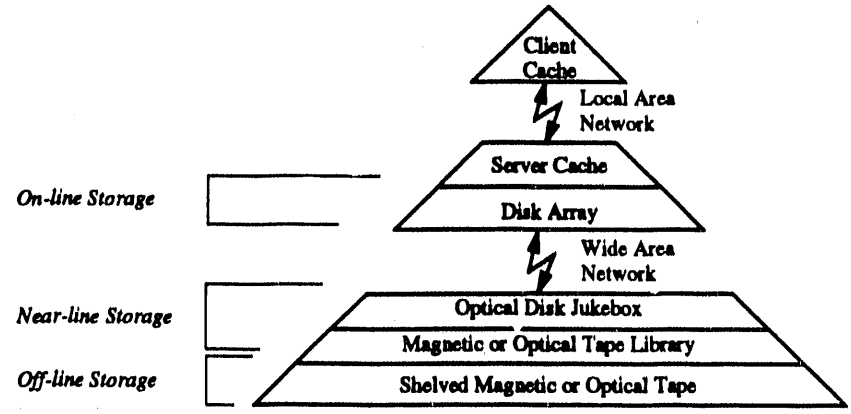

Figure 5
Typical Storage Hierarchy, Circa 1995

Conventional disks have been replaced by disk arrays, a method of obtaining much higher I/O bandwidth by striping data across multiple disks. A new level of storage, "near line", emerges between disk and tape. It provides very high capacity, but at access times measured in seconds.

Disk arrays to replace large disks.

Because of the rapidly decreasing form factor of magnetic disks, it is becoming attractive to replace a small number of large disk drives with very many small drives. The resulting secondary storage system can have much higher capacity since small format drives traditionally obtain the highest areal densities. And since the performance of both large and small disk drives is limited by mechanical delays, it is no surprise that performance can be dramatically improved if the data to be accessed are spread across many disk actuators. Disk arrays provide a method of organizing many disk drives to appear logically as a very reliable single drive of high capacity and high performance [2].

The two most prevalent disk-array organizations are called RAID Level 3 and RAID Level 5 (*RAID = redundant array of inexpensive disks*). Each of these spreads data across a *stripe-set* consisting of n data disks plus an $n+1$st redundancy disk that provides the capability to recover from any single-disk error. The

RAID Level 3 organization provides high bandwidth by providing access to large blocks (for example, a whole track or cylinder) across all of the disks within a stripe set; since all disks are involved in any single access, there can be only one access at a time. Nevertheless, the RAID level 3 organization is well-suited to such high-bandwidth applications as scientific image processing.

By contrast, the RAID Level 5 organization provides a high I/O data rate by providing access to smaller units, even down to the individual sector. This makes it possible to support multiple simultaneous accesses, but at the cost of more frequent updating of the redundancy information.

Near-line storage: not quite on-line, but better than archival.

A comparable revolution has taken place in tertiary storage: the arrival of near-line storage systems. These provide relatively rapid access to enormous amounts of data, frequently stored on removable, easy to handle optical disk or magnetic tape media. This is accomplished by storing the high capacity media on shelves that can be accessed by robotic media "pickers". When a file needs to be accessed, special file management software identifies where it can be found within the tape or optical disk library. The picker exchanges the currently loaded media with the one containing the file to be accessed. This is accomplished within a small number of seconds, without any intervention by human operators. By carefully exploiting caching techniques, in particular using the secondary storage devices as a cache for the near-line store, the very large storage capacity of a tertiary storage system can appear to have access times comparable to magnetic disks at a fraction of the cost.

Optical disk technology for near-line storage.

Optically recorded disks have long been thought to be ideal for filling the near line level of the storage hierarchy [3]. They combine improved storage capacity (2 gigabytes per platter surface originally to more than 6 gigabytes per side today) with access times that are approximately a factor of ten slower than conventional magnetic disks (several hundred milliseconds). The first generation of optical disks were written once, but could be read many times, leading to the term "WORM" to describe the technology. The disk is written by a laser beam. When it

is turned on, it records data in the form of pits or bubbles in a writing layer within the disk. The data are read back by detecting the variations of reflectivity of the disk surface.

The write-once nature of optical storage actually makes it better suited for an archival medium than near-line storage, since it is impossible to accidently overwrite data once it has been written. A problem has been its relatively slow transfer rate, 100K – 200K bytes per second. Newer generations of optical drives now exceed one megabyte per second transfers.

Magneto-optical technologies, based on a combination of optical and magnetic recording techniques, have recently led to the availability of erasable optical drives. The disk is made of a material that becomes more sensitive to magnetic fields at high temperatures. A laser beam is used to selectively heat up the disk surface, and once heated, a small magnetic field is used to record on the surface. Optical techniques are used for reading the disk, by detecting how the laser beam is deflected by different magnetizations of the disk surface. Read transfer rates are comparable to those of conventional magnetic disks. Access times are still slower than a magnetic disk due to the more massive read/write mechanism holding the laser optics, which takes longer to position than the equivalent low mass magnetic read/write head assembly. The write transfer rate is worse in optical disk systems because (1) the disk surface must first be erased before new data can be recorded, and (2) the written data must be reread to verify that it was written correctly to the disk surface. Thus, a write operation could require three disk revolutions before it completes.

Nevertheless, as the form factor and price of optical drives continue to decrease, optical disk libraries are becoming more pervasive. The recent announcement of a consumer-oriented recordable music compact disk could lead to dramatic reductions in the cost of optical disk technology. Currently available systems range from inexpensive systems providing 20 gigabytes in a desk-side unit the size of a three drawer filing cabinet, to full-blown systems

with a maximum capacity of more than 1,000 gigabytes. Transfer rates are measured in hundreds of kilobytes a second at the low end, up to one megabyte per second at the high end, and at both ends of the spectrum the medium change time is about 7 seconds.

Magnetic tape technology for near-line storage

The sequential nature of the access to magnetic tape has traditionally dictated that it be used as the medium for archive. However, the success of automated tape libraries has demonstrated that tape can be used to implement a near-line storage system. The most pervasive magnetic tape technology available today is based on the IBM 3480 half-inch tape cartridge, storing 200 MBytes and providing transfer rates of 3 MBytes per second. However, there has been an enormous increase in tape capacity, driven primarily by helical scan recording methods. In a conventional tape recording system, the tape is pulled across stationary read/write recording heads. Recorded data tracks run in parallel along the length of the tape. On the other hand, helical scan methods slowly move the tape past a rapidly rotating head assembly to achieve a very high tape-to-head speed. The tape is wrapped at an angle around a rotor assembly, yielding densely packed recording tracks running diagonally across the tape. The technology is based on the same tape transport mechanisms developed for video cassette recorders in the VHS and 8mm tape formats and the newer digital audio tape (DAT) systems.

Each of these systems provides a very high storage capacity in a small easy-to-handle cartridge. The small form factor makes these tapes particularly attractive as the basis for automated data libraries. First generation systems, based on the 8mm video tape format, can store 2.3 gigabytes and transfer at approximately 250 kilobytes per second. A second generation system now available doubles both the capacity and the transfer rate. A tape library system based on a 19-in. rack can hold up to four tape readers and over one hundred 8-mm cartridges, thus providing a storage capacity of 250 – 500 gigabytes per second.

DAT provides smaller capacity and bandwidth than 8-mm, but enjoys certain other advantages [4]. Low cost tape readers in the 3.5-in. form factor, the size of a personal computer floppy disk drive, are readily available. This makes possible the construction of tape libraries with a higher ratio of tape readers to tape media, increasing the aggregate bandwidth to the near-line storage system. In addition, the DAT tape formats support subindex fields that can be searched at a speed two hundred times greater than the normal read/write speed. A given file can be found on a DAT in an average search time of only 20 seconds, compared to more than ten minutes for the 8-mm format.

VHS-based tape systems can transfer up to 4 MBytes/second and can hold up to 15 GBytes per cartridge. Tape robotics in use for the broadcast industry have been adapted to provide a near-line storage function.

Helical scan techniques are not limited to consumer applications, but have also been applied for certain instrument recording applications, such as satellite telemetry, that require high capacity and high bandwidth. These tape systems are called DD1 and DD2. A single tape cartridge can hold up to 150 GBytes, and can transfer at a rate of up to 40 MBytes/second. However, such systems are very expensive, and a good rule of thumb is that the tape recorder will cost $100,000 for each 10 MBytes/second of recording bandwidth it can support.

Optical tape technology for near-line storage

A recording technology that appears to be very promising is optical tape [5]. The recording medium is called digital paper, a material constructed from an optically sensitive layer that has been coated onto a substrate similar to magnetic tape. The basic recording technique is similar to write once optical disk storage: a laser beam writes pits in the digital paper to indicate the presence (or absence) of a bit. Since the pits have lower reflectivity than the unwritten tape, a reflected laser beam can be used to detect their presence. One 12-in. by 2400-ft. reel can hold 1 terabyte of data, can be read or written at the rate of 3 megabytes per second, and can be accessed in the remarkable average time of 28 seconds.

Summary

Table 1 summarizes the relevant metrics of the alternative storage technologies, with a special emphasis on helical scan tapes. The metrics displayed are the capacity, bits per inch (BPI), tracks per inch (TPI), areal density (BPI × TPI in millions of bits per square inch), data transfer rate (kilobytes-per-second sustained transfers), and average positioning times. The latter is especially important for evaluating near-line storage media. An access time measured in a small number of seconds begins to make tape technology attractive for near-line storage applications, since the robotic access times tend to dominate the time it takes to pick, load, and access data on near-line storage media.

Technology	Capacity (MB)	BPI	TPI	BPI*TPI (million)	Data xfer (KBytes/s)	Access Time
Conventional Tape						
Reel-to-Reel (1/2")	140	6250	18	0.11	549	minutes
Cartridge (1/4")	150	12000	104	1.25	92	minutes
IBM 3480 (1/2")	200	22860	38.1	0.87	3000	seconds
Helical Scan Tape						
VHS (1/2")	15000	?	?	?	4000	minutes
Video (8mm)	4600	43200	1638	70.56	492	minutes
DAT (4mm)	1300	61000	1870	114.07	183	20 seconds
Optical Tape						
CREO (35mm)	1TB	9336000	24	224	3000	28 seconds
Magnetic Disk						
Seagate Elite (5.25")	1200	33528	1880	63.01	3000	18 ms
IBM 3390 (10.5")	3800	27940	2235	62.44	4250	20 ms
Floppy Disk (3.5")	2	17434	135	2.35	92	1 second
Optical Disk						
CD ROM (3.5")	540	27600	15875	438.15	183	1 second
Sony MO (5.25")	640	24130	18796	453.54	87.5	100 ms
Kodak (14")	3200	21000	14111	296.33	1000	100's ms

Table 1
Relevant Metrics for Alternative Storage Technologies

References

[1] Nelson, M., J. K. Ousterhout, and B. Welch, "Caching in the Sprite Network File System", *A.B.M. Transactions on Computer Systems*, V. 6, N. 1, (February 1988), pp. 134-154.

[2] Katz, R., G. Gibson, and D. Patterson, "Disk System Architectures for High Performance Computing", *Proceedings of the IEEE*, Special Issue on Supercomputing, (December 1989).

[3] Ranade, S. and J. Ng, *Systems Integration for Write-Once Optical Storage* (Meckler, Westport, CT, 1990).

[4] Tan, E., and B. Vermeulen, "Digital Audio Tape for Data Storage", *IEEE Spectrum*, V. 26, N. 10, (October 1989), pp. 34 - 38.

[5] Feder, B. F., "The Best of Tapes and Disks", *New York Times*, Sunday Business Section, (September 1, 1991), p. 9.

3.5.2. Multiprocessor and Parallel Architectures

3.5.2.1. Processor considerations

Because uniprocessor systems are approaching the physical limits of performance, we have begun to investigate various forms of multiprocessor architectures. Three of these are discussed.

The 1990's will see the advent of parallel processing on a grand scale as uniprocessor systems approach their physical limits. The clock speed of single processor supercomputers and reduced-instruction-set CPU's (RISC systems) will reach one nanosecond, but substantially faster speeds may be beyond the physical capability of electronic systems. Scientists requiring hundreds of gigaflops to solve Grand Challenge problems will be forced to use multiple processor systems. However, as long as the development time and cost of parallel programs remains prohibitive, sequential computing will continue to dominate production scientific computing. Today, simply stated, sequential programming still provides the fastest path from program inception to results. To realize *cost-effective* parallel processing, we must supplement high-performance multiple processor systems with the tools to program them (but see section 3.5.2.3 below).

In the next few years, we can expect the emergence of one or two dominant architectures in each of three broad classes of multiprocessor systems: tightly-controlled, loosely-controlled, and cluster. The variety of architectures available now is characteristic of a young field that spawns many different solutions to ensure a

broad search of the solution domain. As the field matures, a few designs will prove superior and manufacturers will rush to develop only those designs. Inferior architectures will either be discarded or carve out special-purpose niches in which their attributes are superior to the dominant, but more general, systems. In each class, dominant trends are already emerging.

Tightly-controlled systems

This class of architectures is defined by distributed-memory SIMD (single instruction stream, multiple data stream) systems and is the most mature of the three classes of systems. The dominant architectural design is a large-number of simple processors interconnected by an extended mesh that supports nearest-neighbor communications. Current systems range from 16,000 4-byte processors to 64,000 1-bit processors, but should grow to 256,000 processors by the middle of the decade. With each processor executing 10-50 megaflops,the largest systems should deliver 2-12 teraflops. The SIMD model, already familiar to users of vector supercomputers, extends naturally to these machines. Although distributed-memory SIMD systems are not suitable for all Grand Challenge applications, they will remain a dominant force in scientific computing. Unresolved issues include: hardware support for floating point operations, I/O, and multiprogramming.

Loosely-controlled systems

This is the broadest and most populated of the three classes. In general, these systems consist of a number of homogeneous processor-memory pairs interconnected by a high-speed communication network, in a MIMD (multiple instruction stream, multiple data stream) configuration. The system's memory is distributed but may support a single, global address space. While still an immature class encompassing many different architectures, interconnection networks, and execution models, some dominant trends are emerging. We expect future systems to consist of up to 16,000 microprocessors interconnected by a two-dimensional mesh of high-speed buses. The combination of miniaturization, short-development time, and simplicity should lead to the dominance of RISC-based microprocessor systems over multiple-chip designs. (RISC processors achieve very high speeds by employing

only very basic instruction sets; the programmable instructions used by the operating system are implemented in terms of these basic sets.) By the mid-1990's, superscalar microprocessors comprised of multiple pipelined functional units will achieve peak execution rates of 500 megaflops. This will allow the largest distributed-memory MIMD systems to deliver between 4 and 8 teraflops. Most systems will use a two-dimensional bus-based mesh interconnection networks in preference to hypercube or multi-stage networks.

To realize the high-execution rates of loosely-controlled systems, applications will require a high computation-to-communication work ratio, and significant amounts of local parallelism. Since we expect buses to reach bandwidths of 100-250 megabytes per second, programs that have significant communication requirements will be bus-bandwidth limited. High levels of parallelism within local tasks will be required to keep the multiple pipelined functional units busy. Providing adequate hardware support and the programming tools to develop such programs will be a major challenge for computer scientists in the next decade. A promising hardware solution is the support of a global address space in which remote memory fetches are satisfied by hardware and the operating system. This solution alleviates the need for explicit messages in application programs, improving both execution speed and programming efficiency.

Clusters

Clusters are stand-alone computer resources, possibly heterogeneous, connected by a local-area network (LAN). Presently, small groups of offices are clustered about a central file server. The offices share the cluster's resources transparently (file server, workstations, printers, and LAN). As the networking of the work environment accelerates over the next five years, an institution's computers will act more like a single machine than interconnected individual actors. The recently announced NREN project should eventually lead to a single national cluster. As the execution speed of workstations increases, the computing power of even small clusters will be enormous. The Supercomputing Resource Center in

Bowie, Maryland estimates that their clusters of workstations are equivalent to 20 Cray-X/MP processors.

Hardware issues here are limited to the development and installation of fast intra- and inter-site networks. The UltraNet, a gigabit intra-site network technology, is now available. Research projects to develop networks with similar data rates based on the HiPPI (high-performance parallel interface) standard are in progress. Inter-site data rates should approach 1 gigabit per second over commercial phone lines in the next five years. The software issues are extensive. To create seamless clusters of heterogeneous machines spread over large geographical areas will require an intensive and coordinated research effort by government agencies, national laboratories, industry, and academia. Users should be able to move files effortlessly from one system to another and visualize data on a variety of media. Jobs should automatically migrate to idle machines and the concurrent tasks within jobs should spread out over the cluster, each executing on the machine best suited to its computational needs. Such a *supercluster* will require standardization of protocols, operating systems, instruction sets, storage systems, data representation, etc. As the systems become better integrated, security will be critical: a renegade process could quickly spread and bring down the nation's entire computing apparatus if adequate security measures are not implemented.

3.5.2.2. Memory Considerations

Distributed or shared memory?

The memory in a multiprocessor can be distributed across the processors or shared. In a distributed memory computer, when a process on one node requires information or data owned by another node, this information must be sent explicitly as a message from one node to the other. The advantage of this method is that it is more flexible. Nodes with different types of processors can be used to adapt to specialized problems. Distributed memory systems are easier to expand, and several smaller problems can be run concurrently without interfering with one another. The disadvantage of the distributed memory system is the overhead involved in passing

messages, especially if there are many data transactions that must be take place between processors.

Shared-memory systems.

In a shared memory system, memory is global and is shared by all of the processors. Parallel computers of this type use one of two general ways to access memory: a common system bus or a switching network.

In a bus-based system, all processors must use the system bus to access the system memory. The bus can be used by only one processor at a time. A bus-based system is efficient up to a point, and frequently good for multiple users whose programs are not large or complex. However the bus is a bottleneck that limits the speed at which processors can access memory. This limits the number of processors that can effectively be added to the system, and therefore also limits the performance of the system as a whole. Typical systems are limited to about ten processors.

The switching network was designed to avoid the memory access bottleneck by providing many different paths to memory while maintaining a single global address space. In this type of system, the processors are connected to memory through one or more layers of switches, the number of layers being directly proportional to the logarithm of the number of processors in the system [1].

While offering greater bandwidth than the bus-based system, the switching network is susceptible to "hot spots". This happens when multiple memory references simultaneously compete for the same switch, thus causing a significant degradation of the whole network. Another problem with the switching network is that memory access is slower, and has nonuniform access time. In addition, a switching network adds significant complexity to the system, affecting system cost, reliability, and scalability of the system. An example of a supercomputer that uses a switching network is the Cray Y-MP.

Shared memory systems are easier to program because the programmer need not worry about subdividing data and allocation parts to individual memories [2]. The potential advantages of a shared memory architecture

have lead DARPA to award a $7.5 million contract to Tera Computer Company to develop a 64-bit, shared-memory parallel computer that will have up to 256 processors. A prototype is expected by 1993.

In practice, there are only three types of parallel machines produced today: MIMD coarse shared, MIMD coarse distributed, and SIMD fine distributed. MIMD machines can make use of either shared or distributed memory, but currently only distributed memory is practical for systems with thousands of processors. MIMD machines must be coarse grain because synchronization costs are too high with fine grains.

3.5.2.3. Software for Parallel Processing

Some software for managing parallel systems is beginning to arrive.

Recently several software packages have emerged that provide the ability to do parallel processing by distributing the work over many machines connected by a network. Packages such as Linda [3], Express [4], and ISIS [5] attempt to make the issue of where the processor is located completely transparent to the programmer or user. These packages provide a parallel operating environment that can be used to handle the coordination of processors on a single MIMD machine, or between processors on several machines, or a combination of these. Linda and Express have been used for parallel processing on networks of Unix workstations, as well as on conventional supercomputers and MIMD machines. Linda uses "tuple space" to maintain data consistency among processes. It allows one to think in terms of shared memory even when no physically shared memory exists. Processes started by Linda can read, extract, or insert tuples, which resemble records in a database. Linda can also provide dynamic load balancing, which is often a problem in MIMD systems [3].

ISIS, developed at Cornell University, is designed to address the issues of distributing parallel processes, as well as the issues of fault-tolerance, automatic software and hardware crash recovery, dynamic reconfigurability, and distributed correctness constraint maintenance [5].

3.5.2.4. Parallel Computer Architectures in the Near Future

We have moved beyond the question of the practicality of parallel processing to the question of how we best put this new tool to work.

The question of the early 1980's was whether parallel computation would become practical. This question has been answered in the affirmative and we now move on to the next set of questions. These questions include the following: What are the best parallel architectures for given classes of problems? How can we best partition a problem into thousands of parts? How do we design algorithms so that delays of interprocessor communication can be kept to a small fraction of the computation time? How do we design the system to keep the load balanced across the available processors? How do we design algorithms that easily scale to use any number of available processors [6]?

SIMD and MIMD, shared memory and distributed memory systems are all proving to be useful ways to design parallel machines. Some algorithms map particularly well to SIMD systems, and others map well onto MIMD systems. In solving many problems, it would be best if parts of the algorithm ran on a fine grained SIMD, while other parts were running on a course grained MIMD. In the future it is probable that scientific supercomputers will combine both SIMD and MIMD in some way. In fact, this is already beginning to take place. At the Pittsburgh Supercomputer Center (PSC), a Cray Y-MP has been connected to a Thinking Machines CM-2 via a HiPPI channel, and data transfers at a rate of about 100Mbits/second are being achieved [7]. Applications have been developed that use this HiPPI connection to achieve speeds considerably faster than could be achieved using either the Cray or the CM-2 alone. Cray is reportedly working on a massively parallel unit that will be accessed as a giant co-processor. Systems such as Linda and Express could conceivably make the use of both types of machines completely transparent. Using one of these methods, supercomputer users will be able to take advantage of the strengths of both SIMD and MIMD systems simultaneously.

3.5.2.5. Gigabit Networks and Supercomputers

Sophisticated networking may change our concept of what a "computer" is, from a locally-connected set of components to a nationally-connected set.

Gigabit-per-second wide-area networks, which operate at speeds equivalent to supercomputer channel speeds, promise to alter the way the networks are used with high-performance computing systems. These networks will allow the creation of "network supercomputers", which are computing systems comprised of geographically distributed components communicating with each other. Such a network supercomputer would be comprised of a number of high performance computing components. These components could include a vector supercomputer, a SIMD parallel supercomputer, a MIMD parallel supercomputer, a very large high-speed file system, a high-resolution graphical display, a source of real-time data, and others.

A researcher will be able to "construct" a supercomputer tailored to his or her specific computation needs by connecting network-attached components. This network supercomputer would "exist" only for the duration of the computations.

Summary

The technology of single-processor supercomputers has nearly reached its theoretical performance limits. To go beyond this limit the next step must be a move to parallel computers. In the coming decade teraflop parallel machines will become available, and if researchers' predictions are correct, many scientific breakthroughs will result. This new breed of massively parallel machines will, in the long run, have an impact as profound as microcomputers did. However, to be able to fully utilize the computing power of these machines, we must rethink our approaches to designing algorithms.

References

[1] S. Ragsdale, "Parallel Programming Primer", Intel Corporation.

[2] L. D. Wittie, "Computer Networks and Distributed Systems", *IEEE Computer*, September, 1991.

[3] D. Gelernter, "Getting the Job Done", *Byte*, November, 1988.

[4] J. Flower and A. Kolawa, ParaSoft/Express information packet, ParaSoft Corporation, 1991.

[5] K. Birman, T. Joseph, and F. Schmuck, *ISIS: A Distributed Programming Environment. User's Guide and Reference Manual* (Cornell University, March 1990).

[6] P. J. Denning and W.F. Tichy, "Highly Parallel Computation", *Science*, November, 1990.

[7] G. Huntoon, Pittsburgh SuperComputer Center, personal communications.

4 Conclusion

Continuing change, but dominated by change in the work rather than change in the tools.

Change is still the order of the day in the ITR arena. As is clear from the examples of Section 3, IT devices will continue to get bigger (in capacity), smaller (in footprint), faster, and cheaper, and the systems that use them will expand in power, flexibility, and usability. New avenues will open as we learn to adapt from serial methods of problem solution to parallel methods; to make more effective use of the image-processing capability of the human mind and the discriminability of the human ear; to create experiential realities to supplement numerical abstractions for the representation of physical processes; and to tap the synergistic advantage of the group over the individual. There are strong commercial pressures to provide the hardware, and strong intellectual pressures to solve the problems. But beyond that, we are going to experience an increased penetration of ITR into the invisible infrastructure of our work environment. It is in this last development that we will begin to see the true fruition of the information revolution.

5 ACKNOWLEDGEMENTS

This ITRA was prepared and edited by David Stevens, Head of the Office of IT Planning of the Lawrence Berkeley Laboratory. Contributing technical specialists, all from LBL unless otherwise noted, were: Barbara Cerny, DOE/RW-12 (*Intelligent Document Processing*), Mark Durst (*Analysis Tools*), Michael Farber, Roy F. Weston, Inc. (*Intelligent Document Processing*), John Feo, LLNL, (*Multiprocessor and Parallel Architectures*), Robert Fink (*Institutional ITR Architecture*), Harvard Holmes (*Institutional ITR Architecture*), Nancy Johnston (*Analysis Tools*), William Johnston (*Institutional ITR Architecture* and *Analysis Tools*), Randy Katz, University of California at Berkeley (*Storage*), Allan Konrad (*The Coming Invisibility of IT* and *Intelligent Document Processing*), Claudia Madison (*Intelligent Document Processing*), Charles McParland (*Integration Tools and Interoperability*), Doron Rotem (*Scientific Data Management*), Arie Shoshani (*Scientific Data Management*), and Brian Tierney (*Multiprocessor and Parallel Architectures*). Any errors of fact or emphasis are the fault of the editor alone.

END

DATE
FILMED
4/15/92