Requirements for Supercomputing in Energy Research:

The Transition to Massively Parallel Computing

RECEIVED

APR 29 1993

OSTI

**U.S. Department of Energy
Office of Energy Research
Office of Scientific Computing
February 1993**

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

MEMORANDUM

Date: February 1993

To: Dr. David B. Nelson
Associate Director for Scientific Computing, Office of Energy Research

Mr. John S. Cavallini,
Deputy Associate Director for Scientific Computing, Office of Energy Research

The Executive Committee of the Energy Research Supercomputer Users' Group (EXERSUG) is pleased to submit to you this documentation of Requirements for Supercomputing in Energy Research. The effort to initiate a comprehensive survey of these requirements was launched on August 17-18, 1992, at the National Energy Research Supercomputer Center (NERSC) in Livermore. On that date, a select group of scientists from all DOE/ER programs met to respond to a request by the Office of Scientific Computing that a study of future requirements and directions for production access to high performance computing resources be undertaken.

It was recognized early that the future in production supercomputing will involve a transition from sequential to parallel computing as high-end computational requirements in the tens of Tera-Flops were identified in a number of ER program areas. Dr. C. William McCurdy of NERSC, and Dr. Stephen T. Elbert of Ames Laboratory accepted the responsibility of preparing and editing the document in response to input and critical review from the DOE/ER technical community. On behalf of all Energy Research supercomputer users, I would like to thank them both for their energy and perseverance in arriving at a composite vision that is broadly endorsed.

Jack A. Byers, Chair
Executive Committee of the Energy Research
Supercomputer Users' Group (EXERSUG)

Executive Committee of the Energy Research Supercomputer Users' Group (EXERSUG)

**Therese Barts, EXERSUG Secretary
Superconducting Super Collider Laboratory
Project Management Office, Accelerator Physics Group
Dallas, TX 75237**

**Chris Bottcher
Physics Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831**

**Jack A. Byers, EXERSUG Chairman
Fusion Energy, M-Division
Lawrence Livermore National Laboratory
Livermore, CA 94550**

**Joe Carlson
Theoretical Division
Los Alamos National Laboratory
Los Alamos, NM 87545**

**Viktor K. Decyk
Department of Physics
University of California
Los Angeles, CA 90024**

**Dennis W. Duke, Interim Director
Supercomputer Computations Research Institute
Florida State University
Tallahassee, FL 32301**

**Thomas H. Dunning, Associate Director
Molecular Science Research Center
Pacific Northwest Laboratories
Richland, WA 99352**

**Judith T. Giarrusso, Deputy Head
Engineering Analysis Division
Princeton Plasma Physics Laboratory
Princeton, NJ 08543**

**William Herrmannsfeldt
Stanford Linear Accelerator Center
Stanford, CA 94309**

Brian E. Hingerty
Office of Health and Safety
Oak Ridge National Laboratory
Oak Ridge, TN 37831

Dale D. Koelling
Division of Material Science
Argonne National Laboratory
Argonne, IL 60439

Jean-Noel Leboeuf
Fusion Energy Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831

Paul Messina, Director
Caltech Concurrent Supercomputer Facilities
Pasadena, CA 91125

Richard A. Nebel
Theoretical Division
Los Alamos National Laboratory
Los Alamos, NM 87545

Michael Ogilvie
Department of Physics
Washington University
St. Louis, MO 63130

Gerald Potter
Program of Climate Model Diagnosis and Intercomparison (PCMDI)
Lawrence Livermore National Laboratory
Livermore, CA 94550

Ned Sauthoff, Head
Physics Department
Princeton Plasma Physics Laboratory
Princeton, NJ 08543

Allan D. Turnbull
Fusion Group
General Atomics, Inc.
San Diego, CA 92121

Table of Contents

<i>Preface</i>	1
<i>Executive Summary</i>	3
I. Introduction: The Emergence of a Practical Path to TeraFlop Computing and Beyond	6
II. Requirements of Energy Research Programs at DOE	11
A. Examples from Key Areas of Energy Research	11
(a) Computational Fluid Dynamics (CFD)	12
(b) Lattice Gauge Theory	14
(c) Electronic Structure, Quantum Chemistry and Chemical Reactions	15
(d) Molecular Biology	16
(e) Plasma Physics (Numerical Tokamak)	17
(f) Climate Modeling (Earth Systems Modeling)	18
B. Summary of Requirements from Application Areas	19
III. Implementation: Supercomputer Production Computing Environment on Massively Parallel Computers	22
A. Introduction	22
B. Initial Production Environment (IPE)	23
(a) Computing Capabilities and Machine Attributes	23
(b) Programming Models	24
(c) Operating System (O/S)	24
(d) Tools	26
(e) Software Libraries	27
(f) Disks and Tertiary Storage	27
C. General Production Environment	29
(a) Operating System	29
(b) Tools	29
(c) Software Libraries	31

IV. Implementation: User Transition to Massively Parallel Computing	32
A. Introduction	32
B. Transitional Issues Facing The Access Center	33
(a) Machine Operation	33
(b) Production Environment	33
C. Transitional Issues Facing the Users.....	35
(a) User Services (Classes and Education)	35
(b) User Services (Interaction with Users)	35
(c) Code Migration	36
V. Recommendations	38
 <i>Appendices: Details of Scientific Requirements</i>	
<i>Appendix A—Computational Fluid Dynamics</i>	<i>41</i>
<i>Appendix B—Lattice Gauge Theory</i>	<i>44</i>
<i>Appendix C—Electronic Structure, Quantum Chemistry and Chemical Reactions.....</i>	<i>50</i>
I. Electronic Structure and Molecular Dynamics	50
II. New Directions for Quantum Monte Carlo Applications	55
<i>Appendix D—Molecular Biology</i>	<i>57</i>
<i>Appendix E—Plasma Physics and the Numerical Tokamak Project</i>	<i>59</i>
I. Numerical Tokamak Project (NTP)	59
(a) Gyrokinetic Particle-in-Cell Methods	60
(b) Fluid Approach to the Numerical Tokamak	61
II. Particle-in-Cell Calculations	64
III. Magnetohydrodynamics (MHD)	65
<i>Appendix F—Climate Modeling</i>	<i>67</i>
I. Earth Systems Modeling	67
II. Spectral Methods for General Circulation Models	70
<i>Glossary of Acronyms and Abbreviations</i>	<i>72</i>

Preface

On August 17 and 18, 1992, a workshop was held at the National Energy Research Supercomputer Center to discuss the requirements for supercomputing in Energy Research programs. Massively parallel computing was immediately identified as the only technology which will fulfill those requirements in the near future, providing a practical path to TeraFlop computing and beyond.

The participants in the workshop were:

Eugene Brooks	Lawrence Livermore National Laboratory
Jack Byers	Chair of EXERSUG, Lawrence Livermore National Laboratory
Phil Colella	U. C. Berkeley
Steve Elbert	Ames Laboratory
Bruce Garrett	Pacific Northwest Laboratory
Dennis Hall	Lawrence Berkeley Laboratory
Bill Herrmannsfeldt	Stanford Linear Accelerator Center
Dennis Hewett	Lawrence Livermore National Laboratory
Brian Hingerty	Oak Ridge National Laboratory
Greg Kilcup	Ohio State University
Michel McCoy	Lawrence Livermore National Laboratory
C. W. McCurdy	Director of NERSC, Lawrence Livermore National Laboratory
Rick Nebel	Los Alamos National Laboratory
Jerry Potter	Lawrence Livermore National Laboratory
Rick Stevens	Argonne National Laboratory

In addition to the above participants, written contributions, or data concerning supercomputing requirements, were received from:

Dan Barnes	Los Alamos National Laboratory
Suse Broyde	New York University
Bruce Cohen	Lawrence Livermore National Laboratory
William Dannevik	Lawrence Livermore National Laboratory
Judy Giarrusso	Princeton Plasma Physics Laboratory
Greg Hammett	Princeton Plasma Physics Laboratory
Gary Kerbel	Lawrence Livermore National Laboratory
Jean-Noel LeBoeuf	Oak Ridge National Laboratory
William A. Lester	U. C. Berkeley
Art Mirin	Lawrence Livermore National Laboratory

This report and its recommendations are based on discussions held at the workshop, consultation with the contributors listed on the preceding page, and discussions with members of the Energy Research Supercomputer Users' Group (ERSUG).

We wish to thank Roxanne Clark for the preparation of contributions during and after the meeting, and the technical production of this report.

C. William McCurdy, Director
National Energy Research Supercomputer Center

Stephen T. Elbert, Computational Scientist
Ames Laboratory, Ames, Iowa

Jack A. Byers, Chair
Executive Committee of the Energy Research
Supercomputer Users' Group (EXERSUG)

Executive Summary

Key Recommendations:

- *The procurement of a massively parallel computer in the Access Program capable of 200-300 billion operations per second for delivery in Third quarter FY94.*
- *Integration of the massively parallel computer with mass storage facilities and an appropriate distributed computing environment.*
- *Research and development of the remaining components of the software environment required for production use of parallel computing.*
- *Integration of the software achievements of the High Performance Computing Research Centers (HPCRCs) in the initial production environment and the use of HPCRC resources to aid in the transition to general use of parallel computing in the Access Program.*

The mission of the Department of Energy's Office of Energy Research is accomplished by programs in Fusion Energy, Basic Energy Sciences, Health and Environmental Research, High Energy and Nuclear Physics, Scientific Computing, and the Superconducting Super Collider. All of these programs depend critically on computational modeling. The application of large-scale computation is necessary for the understanding of fundamental scientific problems in the areas covered by Energy Research programs, and the design of multi-million dollar experiments cannot be done without computer simulation of the experimental apparatus and its components.

This study analyzes the needs of the range of disciplines in Energy Research (ER) in which large-scale computation is a crucial component. The goals are clear in each of these areas, and in most of them good estimates can be made of the computational capability needed to accomplish both near- and long-term milestones. In the cases of climate studies and quantum chromodynamics of elementary particles, for example, the problems are studied by methods which make use of numerical grids. To achieve the needed accuracies, the grids must be made finer than those it is possible to treat with today's computers. In other cases, the interest is focused more on moving from two to three dimensions or additional physics rather than on finer resolution in the models. The ways in which the demands of these calculations scale are understood and provide estimates of the speed and capabilities of the computational hardware necessary to perform them. The scaling of quantum chemical and molecular dynamics calculations,

computational fluid dynamics, and plasma physics, among many other disciplines, are also understood.

From these observations this study concludes that across the spectrum of research and design in Energy Research, many (but not all) critical programmatic goals can be accomplished with a computer capable of approximately 200-300 billion operations per second. It is expected that this machine will be based on a massively parallel computer architecture. Reasonable estimates based on current trends in computational technology place the availability of such a machine sometime in calendar 1994.

The development of massively parallel computers by several manufacturers has progressed from first generation machines to second or third generation machines with progressively better hardware and systems software. The efforts of vendors and those of the DOE High Performance Computing Research Centers (HPCRCs) have, in combination, brought us to a point where it is practical to employ a massively parallel machine capable of delivering 200-300 GigaFlop performance as a production platform. As with most of the early-model supercomputers acquired by the supercomputer Access Program in the past, some work will be required to produce a robust software environment for production. But with new generation machines, it is clear that a massively parallel machine will be an invaluable production platform for Energy Research users, especially in conjunction with current supercomputer resources in the Access Program. Making the first massively parallel production platform available alongside conventional supercomputers will allow users to make the transition from the older to the newer technology in an orderly manner.

This study recommends that such a system be acquired in early 1994. An appropriate timetable to do so is to finalize the requirements for the procurement in the first quarter of calendar 1993, and begin the procurement immediately thereafter. Evaluation of vendors' offerings should be completed by the end of the third quarter of calendar 1993, and the machine and its configuration chosen by that time. Budget authority is required for a third quarter FY94 installation.

Once the vendor is chosen, the Access Program should work intensively with the HPCRCs to move as much as possible of the software environment they have developed to the coming production platform.

This study recommends that a small percentage of the resources of the HPCRCs be set aside to help in the conversion of applications codes to massively parallel architectures. For the larger community of users of the Access Program, the transition to

massively parallel machines is impeded by a combination of lack of access and expert help. Prior to the arrival of the massively parallel machine recommended here, the Access Program should provide aid via consultation and programming assistance to a subset of its users who are granted access to the HPCRCs or other DOE parallel computing facilities.

In a number of ER research disciplines the requirements for the storage of computed data also increase dramatically as research moves to the problems which must be solved over the next three to five years. The entire configuration of a critically needed computational resource for Energy Research is therefore addressed in this study, especially including data storage requirements. By the time of the recommended procurement, it will be possible to satisfy the requirements of the programs by integrating such a machine into the appropriate data storage and distributed computing facilities.

Further conclusions of this report specify the necessary computational software which must be available on such a computer and the services which must be provided to enable workers in Energy Research programs to make the transition to the use of such a massively parallel computer for the majority of their large-scale calculations. The initial software environment should have batch processing and some time-sharing capabilities. Tools for debugging and libraries of application software must be present. Primary data storage on disk must be integrated with mass storage facilities for data migration and archival requirements. All of these features are present in some form today on massively parallel machines from various vendors. By the time of delivery of a system to the Access Program, these software components will have improved further so that, building on the collective experience of the HPCRCs, a satisfactory production software environment can be assembled quickly and continually improved.

Thus, the final recommendation of this report is that research and development for software libraries, programming tools, operating system features, etc. for a parallel computing production environment be undertaken with clear near-term and long-term goals. This effort would be a part of the Software Components and Tools area of the Advanced Software Technology and Algorithms (ASTA) component of the DOE Program Component of the Federal High Performance Computing and Communications Program.

I. Introduction: The Emergence of a Practical Path to TeraFlop Computing and Beyond

A fundamental change is occurring in the technology for supercomputing. Massively parallel computing is emerging rapidly as a practical and necessary tool for scientific computation. The peak capability of the massively parallel computers deliverable by several vendors exceeds, by several fold, the peak capability of conventional vector supercomputers. Rapid advances in software for massively parallel computation are being made by a large number of workers in the area. In addition, a broad spectrum of Energy Research problems is poised to yield to computational resources of the scale this technology offers. Compelling conditions have developed for the provision of access to such resources to the wide community of Energy Research scientists and engineers.

The rapid change in supercomputing technology, as well as the explosive growth of workstation and other distributed computing resources, is driven by advances in microprocessor design and fabrication. For nearly a decade it has been possible to design computers constructed from tens of thousands of slow microprocessors which would, in aggregate, exceed the performance of supercomputers. Microprocessors are no longer slow. Today, the fastest commercially available microprocessors can deliver performance within a factor of five or better of a single central processor of the fastest conventional supercomputer. This development means not only that fewer processors are required to exceed supercomputer performance, but also that a computer program running in parallel on many processors does not reach an intolerable bottleneck if for some period it is able to use only one processor.

For several years, advances in microprocessor technology have allowed the construction of parallel computers whose *hardware* capability exceeds that of conventional supercomputers by several fold. The following diagram suggests that fact.

Based on a sampling of manufacturers' claimed peak speeds. Massively parallel estimates based on 1000-node machines or their equivalents. Projections based on manufacturers' estimates.

In addition, these advances in microprocessors have been accompanied by increased capacity and decreased cost of memory for them. Consequently, the memory sizes of massively parallel machines have grown faster than those of conventional supercomputers, as the diagram below shows.

Massively parallel estimates based on 1000-node machines or their equivalents. Projections based on manufacturers' estimates.

If these trends in hardware capability told the complete story, there would have been no doubt at the end of the last decade that the right move for high performance computing resources was to convert immediately to using massively parallel computers. There are several reasons why the transition to parallel computing has proceeded more slowly.

Parallel computing involves not only the division of a program into computational units which can be attacked separately with appropriate synchronization, but must also take into account the delays caused by the communication between processors. These delays involve multiple levels of potential communication inefficiencies including: latency for data to begin to arrive at another processor, bandwidth for the complete transmission across the network, and the bandwidth and latency to local memory for each processor. Therefore, a perfectly parallel program might still run inefficiently if any of the critical steps associated with moving data from one processor to another is too slow. Faster and more efficient communications networks connecting the processors of massively parallel computers have begun to emerge in recent years, and more improvements are on the horizon. Nonetheless, avoiding bottlenecks associated with communication between processors continues to be a primary challenge for software development on these machines.

One of the principal reasons that massively parallel computers did not capture the majority of scientific computation immediately upon their arrival upon the scene is that the majority of scientists and engineers could not attain supercomputer performance on them because of the barrier presented by software development. Putting aside for the moment the problems of operating system software, compilers and so forth, the foremost problem for scientists has been the development of

parallel algorithms for their respective disciplines. The primary effort in the transition to parallel computation is the "parallelization" of existing user codes.

Additional software problems hampering conversion to parallel machines concern the operating system and coding environment. Advances are being made in these areas as well, and several of the vendors of massively parallel machines which are being delivered in 1992-1993 advertise acceptably mature operating system, debugging and scheduling software. In some cases, this is second or later generation software running on new machines from a particular vendor. In section III of this report we will discuss the software necessary for a production environment on a massively parallel computer.

Over the past two or three years, we have seen remarkable advances in a broad spectrum of areas in the parallelization of application codes. Fortunately, it is not necessary to start from scratch with every code. Not only have libraries of parallelized routines for common tasks, like solving systems of linear equations, been developed, but a much better understanding of techniques and strategies for parallelization has evolved. Some examples help to prove this point.

Quantum Chemistry Code on the Intel Delta Supercomputer

In the area of quantum chemistry, and its application to elucidating materials properties and chemical reactions, a key step is constructing an approximation to the electronic structure of the

molecule(s) in question, called the Hartree-Fock wave function. It has been known for some time that part of that calculation is naturally and easily parallelizable, namely the construction of a large set of integrals necessary for the task. A pacing step which was not generally understood to be obviously parallelizable was the building of the so-called "Fock matrix." The figure on the preceding page shows the performance gained by parallelization of that difficult step, essentially completing the parallelization of Hartree-Fock calculations, which constitute a ubiquitous application in areas as diverse as material science and molecular biology.

The figure also hints at the scaling problem which often lies at the heart of parallel computing performance. Parallel machines are better for large problems than small ones. At some point it is better to increase the size of the task than to apply more processors to solving a small problem.

Another example is provided by the celebrated success of Quantum Chromodynamics (QCD) calculations on parallel machines. In this case, the calculations, known as lattice gauge theory calculations, are conveniently performed using Monte Carlo and sparse matrix algorithms. These calculations were an early success story in parallel computing, because the underlying physics itself is parallel. Furthermore, because the calculations in lattice gauge theory can tolerate a rather large ratio of computation to communication, they were able to run on the early parallel machines (such as the Caltech Cosmic Cube) despite their relatively inefficient networks. Also, the overall size of lattice QCD programs is not large, typically a few thousand lines, and it is usually feasible to hand code the inner loops to optimize performance of both computation and communication. Nonetheless, this example indicates the level of success possible in the wide class of applications which use Monte Carlo algorithms.

Until 1992, machines built by several lattice QCD research groups, a couple of which were capable of sustained performance of more than 6 GigaFlops, have been the dominant computing resources for QCD on the international horizon. In 1992 commercial machines caught up to this level of performance, and the newest offerings will exceed it. There are lattice QCD research groups with exclusive or dominant use of CM-2s from Thinking Machines, as well as i-860-based machines such as Intel's Delta and Meiko's Computing Surface. Performance on these machines is competitive with the custom built machines, for example, lattice QCD can sustain about 6 GigaFlops on a full 64K node CM-2.

It has become clear that the path to TeraFlop computing and beyond lies in the use of massively parallel computers. The argument is made forcefully by an examination of the trends in computing hardware development. It is, in fact, entirely likely that new generations of conventional (vector) supercomputers may no longer be the dominant engines of high performance computing in the 1995-97 time frame.

But the most compelling arguments for making massively parallel computing resources accessible to all of Energy Research are not based on hardware developments. They are based on the new science which will emerge from the application of these greatly enhanced computational resources. The trends in the basic science disciplines underlying all of the Energy Research programs are in the direction of exploiting massively parallel computation to gain a factor of 1000 or more in capability over the next five years. This trend can be seen in university research, in the computational science funded by the National Science Foundation, in computational aids to

exploration employed by the oil industry, and in the computational fluid dynamics areas pursued by the National Aeronautics and Space Administration. These developments are of direct concern because of their effect on the pool of computational science talent available to DOE. But the trend toward parallel computation is also evident in the Department of Energy where the establishment of the High Performance Computing Research Centers (HPCRCs) is a major step in the transition of DOE computational science to massively parallel machines.

It is time to establish a strategy for the transition of computational science in the DOE to this new technology. This report describes a key element of that strategy, namely the provision of parallel computing resources to Energy Research scientists and engineers through the Access Program. This effort is a natural and necessary follow-on to the establishment of the HPCRCs.

In the discussions on which this report is based, there was no question about whether the correct path for computationally intensive science in Energy Research is the one to massively parallel computing. The bulk of Section II of this report is dedicated to an analysis of the requirements of specific programs and disciplines in Energy Research. Further details on mission requirements for massively parallel computing are given in the appendices.

The remainder of this report is organized to exhibit the requirements for massively parallel supercomputing and to evaluate the technologies and strategies necessary to establish a production facility. Section II describes the requirements for Energy Research programs with specific examples. The production computing environment and its evolution are described in Section III. The effort necessary on the part of users and on their behalf to effect the transition is discussed in Section IV. The final recommendations are given in detail in Section V.

II. Requirements of Energy Research Programs at DOE

A. Examples from Key Areas of Energy Research

High Performance Computing is critical to DOE's Energy Research mission. To assess requirements in high performance computing for DOE Energy Research, a workshop was held in August of 1992. The attendees represented key areas of Energy Research programs. Research in the programs of Fusion Energy, Basic Energy Sciences, Health and Environmental Research, High Energy and Nuclear Physics, Applied Mathematical Sciences and the Superconducting Super Collider depends heavily on computational modeling. The application of high performance computing in these programs ranges from the modeling of the entire globe for climate studies to the quantum description of new materials on the atomic scale. While it would not be possible to outline the requirements for every area of computational science which is critical to the DOE Energy Research mission, it is easily possible to describe some of the most important and visible applications of high performance computing and to characterize their requirements. The participants in the workshop were chosen from disciplines spanning the range of DOE programs.

This section summarizes the computational requirements which grew out of this workshop, in which the needs of a range of typical calculations were analyzed. The areas discussed with and among users and experts in the preparation of this report can be grouped under the following headings, each of which encompasses several distinct subdisciplines funded by Energy Research programs.

- (a) Computational Fluid Dynamics (CFD)
- (b) Lattice Gauge Theory
- (c) Electronic Structure, Quantum Chemistry and Chemical Reactions
- (d) Molecular Biology
- (e) Plasma Physics and the Numerical Tokamak Project
- (f) Climate Modeling

These headings include some of what are called the "Grand Challenges" of computational science as well as other basic problems which will occupy researchers well into the next century. The requirements we outline here focus on some problems that will interest scientists in the next five years. Indeed, while the Grand Challenges, in their entirety, are hardly about to fall under the assault of the high performance computing of the 1990s, many aspects of these problems are computationally accessible.

The scientific areas discussed in this section are not just important problems in basic science. They are examples of efforts in computational science which are critical to DOE's Energy Research mission. It is hardly possible here to specify all the ways in which these efforts are

central to the Energy Research mission, but a few examples give an indication of the importance of scientific computation to that mission. For instance, among its other DOE applications, computational fluid dynamics is a key to understanding the complex flow of burning hydrocarbons and to designing more efficient internal combustion engines. Quantum chemistry and the theoretical study of chemical reactions in porous media are being applied in a large-scale effort to design molecular level strategies to address the environmental remediation problems at DOE facilities. Molecular biology plays an important role in those environmental remediation strategies and is a component of most of the DOE's mission in health and the environment. Plasma physics is the central issue for the design of magnetic confinement fusion energy sources. Climate modeling addresses the issues of environmental change and the impact of energy-related atmospheric emissions. Lattice gauge theory provides the scientific guidance to accelerator experiments such as the Superconducting Super Collider. A complete list of examples of computational science which are critical to the Energy Research mission would form a book of its own.

The purpose of this section is to specify the requirements for making major progress in these areas. This report is not concerned with incremental increases in resolution, or the addition of some ancillary physics within an existing model, but with qualitative leaps in the fidelity of the models in representing the physical phenomena studied. The possibility of this kind of progress is based on the fact that machines being offered towards the beginning of this time frame will advertise peak speeds and memories about two orders of magnitude greater than that of the Cray C90 (single processor mode). A superficial observation tells much of the story: two orders of magnitude translates to a transition from routine 2-D to routine 3-D modeling for many problems, or at fixed dimensionality, a transition from poorly resolved problems with limited physics to highly resolved calculations with complete physics.

(a) Computational Fluid Dynamics (CFD)

Computational Fluid Dynamics (CFD) stands to be one of the primary beneficiaries of the push to massive parallelism. The availability of the computational capacities discussed above, combined with the advances in numerical algorithm technology achieved over the last fifteen years, will enable us to analyze and understand fluid flows for problems of substantial scientific and technological importance. For example, in power plants and automobile engines, the efficiency with which the burning of hydrocarbon fuels produces useful energy, as well as the rate at which pollutants are produced is strongly dependent on the detailed interaction between the burning and the fluid-dynamic mixing of the fuel, the air, and the combustion products. In the design of computer components, fluid dynamics drives the cooling of circuits, and is a controlling factor in the design of high-speed disk drives.

All of these applications are 3-D and time-dependent, have complex physical processes coupled to the fluid dynamics, and have complex physical geometries. On the present generation of conventional supercomputers, we are unable to compute problems of this complexity. A typical state-of-the-art CFD code running on a conventional supercomputer can produce accurate solutions to one of two classes of problems: either very simple physics models in complex 3-D geometries, or complex physics models in simple, primarily 2-D geometries.

A significant step in the modeling of fluids will have been taken when it will be possible to routinely model fully resolved problems in three dimensions instead of two. A fully-resolved 2-D calculation performed on a uniform finite-difference grid typically requires a 1000 x 1000 grid in two dimensions, and can be done on the current generation of supercomputers. Thus, a comparably-resolved 3-D calculation would require at least an 1000 x 1000 x 1000 grid, a factor of 1000 more computation. Questions that must be addressed are to what extent that the numerical methods under consideration can serve to decrease these computational requirements and to what extent are they applicable to massively parallel architectures.

The answers to both of these questions lie in the detailed mathematical properties of the problems being solved. The first of these is locality: disjoint regions in space are nearly decoupled. Locality is a general property of all of the equations of fluid dynamics, inasmuch as they are described by partial differential equations of the three classical types: elliptic, parabolic and hyperbolic. The second property of the equations is intermittency: that typical solutions consist of small regions where the solution gradients are large, imbedded in regions of smooth flow.

The desire to exploit intermittency has led the designers of CFD algorithms in recent years to consider adaptive algorithms, i.e., algorithms that concentrate computational degrees of freedom in the regions containing large gradients and reduce the number of computational degrees of freedom where the gradients are small. In adaptive mesh refinement, for example, mesh points in a finite difference calculation are inserted and deleted dynamically as the calculation proceeds in response to the changing conditions in the flow. In two dimensions, the use of such algorithms have been shown to reduce the cost of calculations by a factor of 10 or more; preliminary results indicate that this will continue to hold in three space dimensions. Thus, we expect that with the use of adaptive algorithms only a factor of 100 in additional computational power is required to obtain the breakthrough-level results desired for the problems described above.

In a similar fashion, the locality properties of the equations of fluid dynamics allow us to map these calculations very effectively onto massively parallel architectures. Locality leads to numerical algorithms in which the bulk of the computational effort is used to compute the dependence of the solution at a point at nearby points in space. Thus a natural strategy for parallelizing CFD calculations having these properties is to distribute spatially disjoint regions on to different processors.

Thus, the presence of a massively parallel computer would allow CFD to make a substantial impact on a variety of problems of substantial importance to DOE. However, the way in which it will make this impact is by allowing the development of a whole new generation of simulation codes with qualitatively different capabilities, rather than simply scaling up existing codes by adding more grid points. Thus, for a massively parallel machine to be useful in achieving this goal, it must constitute an appropriate environment for developing such advanced software. It must provide appropriate languages and tools for expressing the programmer's intent regarding parallel execution and data layout that are sufficiently general to support the kinds of dynamically-changing data structures represented by the adaptive methods discussed here. These calculations will generate enormous amounts of data (up to 16 GigaBytes/time step, or 10^{13} words per run). We must be able to interactively browse this data, in order to extract useful information from the calculations.

(b) Lattice Gauge Theory

Lattice gauge theory is a subfield of theoretical particle physics which uses supercomputers to study quantum field theories. With only a slight over generalization, one might identify two classes of problems which are solved on the supercomputer. The first comprise relatively small-scale investigations of points of principle, perhaps to provide qualitative understanding of a particular theory, or to see if it is feasible to compute some quantity, or to study a new algorithm. The second class of problems comprise “production runs” aimed at the reliable extraction of quantitative predictions which may be of relevance to experiment. These are almost exclusively performed in the context of lattice quantum chromodynamics (QCD), the lattice field theory believed to describe the interactions of quarks and gluons. A competitive calculation usually requires several thousand Cray hours. Typical outputs include: the masses of the proton and other hadrons, studies of hadronic structure, matrix elements of weak interaction operators, and thermodynamic parameters of the transition to the quark-gluon plasma. For several of these quantities, notably weak interaction matrix elements and the chiral phase transition temperature, lattice QCD calculations are accurate enough to play a significant role in the confrontation of theory and experiment.

There are five parameters which govern the quality of the results of lattice QCD runs: (1) the number of points in the 4-D grid, (2) the resolution of the grid, i.e., the physical spacing between adjacent lattice sites, (3) the quark masses, (4) the statistics of the Monte Carlo sample, and (5) whether or not one includes the effects of vacuum polarization. No simulation at present—or in the next few years—can work in a regime where all five control parameters are in their physical or asymptotic range. Thus, some extrapolation is always required to extract a physical answer. Of these five parameters, the one that costs the most to control is the inclusion of vacuum polarization. Simulations which make the uncontrolled approximation of ignoring this effect (quenched simulations) require about two orders of magnitude less computer time than those in the correct theory (dynamical simulations). Current (1992) state-of-the-art calculations are performed on 4-D grids of size up to 32^4 with typical lattice spacings of about 0.05 Fermi. Those which use the quenched approximation can be run on traditional vector supercomputers; those using the full theory with dynamical fermions are currently run almost exclusively on massively parallel machines.

As discussed above, to get a physical answer from a lattice QCD calculation, one must at present extrapolate in three parameters: lattice spacing, lattice volume, and quark mass. In addition, although one may use the “quenched approximation” in a first round of calculations, in the end, one should repeat the calculation using the full, expensive, theory with dynamical quarks. Each of these effects leads to a systematic error, and even a TeraFlop-year of computing would not be enough to work in the physical limit of this parameter space. Therefore, one must decide how to spend limited computing resources, depending on which of these sources of error is the most important. This decision will depend on the particular physical quantity one wants to compute.

We present only one example here; the prototypical example of a lattice QCD calculation is the hadron spectrum, i.e., the masses of the proton, rho, pion and other related particles. In this case one of course knows the answers nature has given. Thus, a calculation gives a test of the QCD theory, and of lattice QCD in particular. It is probable that no one seriously doubts that this test will be passed, but doing so will give added confidence in other lattice QCD calculations, where the answer is not known.

Figure B1 (in Appendix B) shows some typical recent data concerning the hadron spectrum. The plot shows proton mass versus pion mass, where both are measured in units of ρ meson mass. In the unphysical limit where all quarks become infinitely heavy, we get the point at the upper right of the plot. As the quark mass decreases to the physical point where $m_\pi/m_\rho = 0.18$, one hopes the data will converge towards the physical value of the nucleon mass, marked in the lower left corner. Plainly it is plausible that lattice QCD will correctly reproduce the correct value of the proton mass, but to check we must decrease the quark mass until $m_\pi/m_\rho = 0.18$.

To avoid severe finite volume effects, as one decreases quark mass, one must at the same time increase the number of points in the lattice. One can estimate that to perform the definitive calculation of the hadron spectrum one will have to go from current lattice size (about $33^3 \times 64$) to sizes of about $128^3 \times 256$. Counting flops and bytes, the resources needed for such a calculation are about 300 GigaBytes of main memory, and about 1000 TeraFlop-hours of runtime. Such a definitive calculation would be feasible on a TeraFlop platform. However, in the interim, for purposes of algorithm analysis and to provide intermediate evidence of verification of principle, a calculation on a $64^3 \times 128$ lattice would be most useful. This would require about 30 GigaBytes of memory and 30 TeraFlop-hours. This latter calculation would represent a long, but achievable calculation on a 200-400 GigaFlop computer.

(c) **Electronic Structure, Quantum Chemistry and Chemical Reactions**

There is a wide range of applications of the techniques of theoretical chemistry, and in this summary we focus on only two generic types of problems. A much larger collection of examples and an analysis of the status of the discipline and its future requirements are given in Appendix C.

Electronic Structure: At the heart of theoretical chemistry, which provides the fundamental tools to address the practical problems of the properties of biological molecules, materials, atmospheric reaction rates, solutions of environmental pollutants, and other chemical systems, is the calculation of the electronic structure of molecules. The forces between two reacting molecules, or between a molecule and a solid surface are determined by the quantum mechanical behavior of the electrons which form the bonds of molecules and determine their chemical and physical properties.

As an example, consider a Hartree-Fock calculation of the total energy of a molecule. For Hartree-Fock calculations the scaling is determined by the integral calculations and assembly of the Fock matrix which nominally scale like N^4 , where N is proportional to the number of electrons. Once the molecule becomes large enough (approximately 100 atoms) many of the atoms in the molecule are sufficiently distant from one another that the integrals can either be neglected (set to zero) or approximated by multipole expansions. In this case the scaling becomes approximately N^3 . It is currently possible to do calculations on systems of the order of 100 atoms or so. Periodic Hartree-Fock calculations of total energy for a single geometry on mineral surfaces currently require over 15 hours of CPU time, 128 million words of memory and over 5 GigaBytes of disk storage on the Cray Y-MP at NERSC.

To perform calculations on medium sized fullerenes, or realistic calculations of a mineral plus an organic adsorbate molecule, or calculations relevant to the liquid-surface interface for a mineral requires the ability to treat between 500 and 1000 atoms. Based on the N^3 scaling behavior, such calculations require sustained speeds of between 250 and 500 GigaFlops.

Molecular Dynamics: The simulation of the dynamics of larger systems such as biomolecules or smaller molecules in condensed phases relies on solving classical equations of motion—the Molecular Dynamics (MD) method. The computational effort in these calculations is determined largely by the complexity of the analytic functions used to represent the potential energy surface. MD simulations of biomolecules use potential functions that represent bonded interactions involving pairs, triples, and quadruples of adjacent atoms in the molecule and sums over interactions between pairs of non-bonded atoms. For large enough systems, the non-bonded interactions are the most costly and scale like N^2 , where N is the number of atoms in the simulation. Most MD simulations of liquids are limited to pair-wise interactions. Important electrostatic interactions are typically incorporated by including coulombic interactions between pairs of fixed charges at sites on the ion and in the water molecules

Calculations are possible on current vector supercomputers or on earlier generation parallel computers which include of the order of 9000 atoms. The need at this point is to be able to do calculations on systems which involve, for example, one or two biologically active molecules and the molecules of the solvent in which they move. Such calculations involve roughly 50,000 atoms, an increase of roughly a factor of 50. We also need to be able to do those simulations for longer times than are currently possible.

The requirements for MD simulations are for calculations on systems much larger than those addressed to date, and much longer time scales. The theoretical maximum size of system that can be addressed by the different types of calculations will be determined by how the problem scales. The calculation of pairwise interactions nominally scales like N^2 , but for sufficiently large systems (10,000-100,000 atoms depending on the form of the force field) the scaling decreases from N^2 to N . The physical times treated in the simulation scale linearly with computer time.

With these scaling assumptions, one concludes that 200 GigaFlops will allow significant calculations on reacting, biologically active molecules with adequately treated solvent, while 500 GigaFlops will begin to make these calculations routine.

(d) Molecular Biology

An outstanding problem in the arena of nucleic acid and protein structure determination is the reliable computational location of the static minimum free energy structure in the natural environment of the cell, i.e., in aqueous milieu in the presence of appropriate concentrations of salts. One computational approach is energy minimization. Even though energy minimization yields only static structures, this method is currently severely hampered by lack of adequate computing power.

This technique suffers from the well-known multiple minimum problem and strategies to cope with it have been devised. One such method, known as the build-up technique, has been

successfully employed in the prediction of polypeptide structures. The technique requires global scale searches of the conformation space of the amino acid building blocks, followed by combinatorial build-up of the building block energy minima to generate larger structures a residue at a time. Indeed, current constraints limit the number of duplex residues that can be built combinatorially to three. Furthermore, the energy calculation time increases as the square of the number of atoms, while the minimization increases at a rate greater than linear in the number of variables. This means that the calculation increases at a rate greater than N^3 where N is the number of atoms.) Typically the largest structures tried are 12 double stranded residues of DNA. although extension to 24 double stranded residues is contemplatable on current machines. The search strategies (an entirely separate problem) are very suitable to parallel machines, since many thousands of identical trials can be run in parallel with different sets.

Treating the 146 base pairs of nucleosomal DNA would be an important milestone toward computational structural determination of nucleic acids. That problem has six times the number of base pairs which can currently be treated on conventional supercomputers. Since the problem scales like N^3 , and current efforts make use of a maximum of 1 GigaFlop peak speed, a machine with a peak speed in the neighborhood of 200-500 GigaFlops would be needed to solve this problem in the same time that smaller problems are solved on the 1 GigaFlop conventional machines.

(e) Plasma Physics (Numerical Tokamak)

The Numerical Tokamak project represents one of the most ambitious thrust areas in computational plasma physics today. The initial focus is on developing an understanding of the turbulent energy and particle loss mechanisms in tokamaks, but will expand in time into a program to model the entire device, hence the name, Numerical Tokamak. The effort is divided into two principal areas, particle modeling (gyrokinetic, Particle-in-Cell (PIC)) and the fluid approach.

Current gyrokinetic PIC codes running on Crays utilize gross simplifications of the physics (ranging from artificially large gradients to extremely crude meshes) to realize solutions in 100 hours or less. It is estimated that realistic parameters for these problems would require at least a 1000 fold increase in throughput. A problem which has been out of reach has been a physically realistic representation of global tokamak operation taking into account turbulent processes on transport (long) time scales.

One approach to the problem of global tokamak modeling would be to run the PIC code just long enough to resolve the turbulence, and generate the relevant diffusion coefficients or fluxes. Then associated 1- and 2-D transport codes would utilize these fluxes to advance the entire system in time. Additional modules could model heating mechanisms, such as beams or RF. Only when profiles change significantly would the gyrokinetic calculation be called to generate new fluxes.

The question then comes down to what computational power is required to run this hybrid code. For this code to be useful, the most time consuming portion of the calculation (the generation of the fluxes or diffusion coefficients) must be accomplished in minutes or at most a significant fraction of an hour. This capability is required because the diffusion coefficients will have to be recomputed frequently. Two simplifications can be made which can reduce the computer time requirements while continuing to extract useful physics: (1) the utilization of implicit techniques

permitting far larger time steps and (2) the development of linearized models (which are realistic on the shortened time scale required for diffusion coefficient generation). With these assumptions, it is estimated that a computational resource generating on the order of 200-500 GigaFlops would realize these fluxes in less than one half hour, permitting the use of such a hybrid code. This would represent a significant new development in computational plasma physics.

The fluid approach to plasma dynamics evolves a set of nonlinear moment equations for macroscopic plasma variables. While current computer technology and numerical algorithms are deemed adequate to perform turbulence calculations using fluid models for edge plasma parameters, the next generation of massively parallel mainframes with near-TeraFlop performance, combined with new algorithms and software development, are needed to tackle the fluctuations at the core of present-day tokamaks. The so-called gyrofluid models have gone a long way toward alleviating some of the analytical difficulties.

Parallel versions of gyrofluid codes are highly evolved and have been targeted at a variety of existing parallel computers. Recent performance studies for one of these codes on a 128-node iPSC/860 show that varying the number of modes retained in the fluid calculations changes the optimal number of processors for comparable timing per step. Not surprisingly, more modes entail a greater optimal number of processors. This scalability has been confirmed on the Caltech Intel Delta, where use of all 512 processors has resulted in an order of magnitude improvement over the Cray-2.

While these measures of performance are encouraging, the largest calculations performed to date have involved 1000 radial grid points and 512 modes, in part because of the limitation to 16 MegaBytes of memory per processor. These calculations are a factor of four shy of the minimum resolution required, with 1000 grid points and 2000 modes, to model the fluctuations at the core of the TEXT and TFTR tokamaks. What is required to carry out such minimum resolution calculations routinely would be a computer in the 200 GigaFlop range or more, with at least 128 MegaBytes of memory per processor. A conservative extrapolation of iPSC/860 and Delta scalings to a 1000 processor MPP indicates that a peak speed of 200 GigaFlops would probably be sufficient to model the core of these tokamaks.

(f) Climate Modeling (Earth Systems Modeling)

Modeling plays a vital role in earth systems science, in part because the scope and complexity of the earth system renders it inaccessible to controlled experimental manipulation. A better understanding of earth system behavior is needed as one of the important inputs to comprehensive energy, environmental, and economic policy formulation

At present, we have models of some of the major components of the earth system, such as atmospheric and oceanic dynamics and thermodynamics. Other components are under development. For the most part, the existing component models have been developed independently and in relative isolation. What is required for comprehensive earth system modeling capability is the full coupling of all major component subsystems to properly reflect the realistic interaction among them.

As an example, current-generation atmospheric general circulation models (AGCMs) typically have a horizontal resolution of about 4 deg x 5 deg, and are integrated for a few decades of simulated time with a time step size on the order of a few minutes. This typically requires on the order of 200 CPU hours on vector supercomputers. The desired resolution for this system is actually on the order of 1 deg x 1 deg. What is more, when such a model is coupled to other earth system components with longer time scales (e.g., ocean and terrestrial ecosystem modules), the desired simulation time scale becomes on the order of centuries. To enhance resolution, increase simulation duration, incorporate refined process representations (e.g., improved cloud-radiation interaction), and fully couple all major earth system components into the next-generation model clearly will require TeraFlop performance levels.

However, before such a computation can even be designed, much can and must be learned about the behavior of coupled versions of current-generation models. What is desired is to be able to undertake pioneering experiments involving the fully-coupled earth system model, constructed by linking present-generation subsystems (atmospheric and oceanic GCMs, newly-developed global ecosystem dynamics models, and biogeochemistry models). This is an essential step on the path to the next-generation simulation capability, because the increased knowledge gained from experiments with the coupled system will provide guidance on how best to improve the fidelity of process representations and spatial resolution in each component. To model coupled problems in this experimental context would require peak speeds on the order of 300-500 GigaFlops.

For example, early coupled model experiments with coarse grids will shed light on the long-term natural variability of the ocean-atmosphere-land-biosphere system, and on the memory time of the system for initial conditions in the "slow" components. Multiple calculations with such models must be executed to determine the sensitivity of the long-term behavior of the system to initial conditions in deep ocean circulation and biomass geographical distribution.

Finally, it should be noted that the data volumes associated with these types of experiments are truly enormous. We are dealing with the statistical properties of 3-D coupled chaotic dynamical systems with widely disparate space and time scales, integrated in time with essentially explicit time advance algorithms. The need for advanced data handling and scientific visualization facilities is perhaps nowhere more acute, among the computational grand challenges, than in comprehensive earth systems modeling.

B. Summary of Requirements from Application Areas

An argument common to all the application areas is that the current generation of vector supercomputers is only able to nibble at the edges of Grand Challenge computational problems. Most of the analyses conclude that 100-500 GigaFlop computers are required to allow substantive enhancements in physics models, including more complex geometries involving higher dimensionality, refined physics models, or integration in time for extraordinarily long periods. This conclusion is clearly expressed, for example, in the previous section analyzing the requirements of computational fluid dynamics.

Application areas such as climatology or computational plasma physics or CFD have a second area of emphasis. In these fields the question posed and the answer expected are strongly dependent on the methods and models used to simulate enormously reduced versions of the actual system studied. As such, the development of successful computational models for these reduced systems comprises a large part of the research activity, and the comprehension of the computational results provides a challenge which in some ways is as great as that posed by the actual simulation. Consequently, there is a strong additional emphasis in these areas on the other service components, such as tertiary storage, and powerful graphical post processing integrated into the computational environment. This emphasis is not found in areas such as QCD or *ab initio* molecular dynamics in which all the physics is embodied in the model, and the major interest is simply to integrate long enough and with sufficient resolution to obtain the accuracy required to verify the physics used in the models. An examination of applications requirements reveals the need for a plan to satisfy them beyond simply providing access to computing hardware.

The computational requirements of ER programs will be addressed in an era of rapidly evolving technologies for massively parallel computing. The evolution of these machines is so rapid that exciting new intermediate results will become accessible at any point at which they are applied in ER research programs. Given the anticipated increases in computing power, it is necessary to understand that a viable computational scientific environment consists of a balanced, yet evolutionary, production environment. At any point in time, there must be a balance between the capabilities of the MP platform (including its tools and software), the scientific applications run or developed, and the support hardware and software, including tertiary storage, workstations, and graphical postprocessors. It is this balance which permits the scientist or engineer to move forward. The whole issue of balance is brought out especially clearly in the Appendices dedicated to climatology and CFD.

Developing this balance will require careful planning. For example, it takes from three to twenty-four months to transform a large sequential code into an efficient parallel code. Even in a mature, balanced MP production environment with adequate storage, post processing capabilities and the most refined MP toolsets, scientific production runs will not happen unless highly parallelized applications codes are there too. Equally important, without a center staff with mastery of the MP environment from systems to storage to tools, there would be no way to effectively utilize the infrastructure. But balance can be combined with rapid evolution as it has been in the Access Program in the past. For example, an early production environment is probably a very good place to develop the early production codes and the critical human infrastructure.

An obvious difficulty is that NERSC users are currently faced with a single, gigantic discontinuous jump. Codes developed on the current generation of supercomputer, the Cray-2 or the C90, are not directly portable. What is more easily portable would be codes developed on early generation MPPs such as the Intel Delta, the CM-2, the BBN TC-2000, the NCUBE, or codes developed on current generation computers at the HPCRCs. While a fair number of codes will be ready for the transition due to the existence of these MP computers, the bulk of the ER codes must still undergo the process of parallelization.

The Access Program has dealt with users' problems in transitions to new machines and software environments in the past. The new problems are more difficult, but now there are additional resources to make the process easier, in particular High Performance Computing Research Centers.

With help from the NERSC staff, users willing to begin modification of production codes for a parallel environment would benefit greatly from access to HPCRC computing resources. A percentage of cycles on those machines could be dedicated to such development. That time could be managed by the current Access Program. Help from NERSC consultants and aid from its programmers could be directed toward helping users in the Access Program make the first steps to parallel computing. Another benefit of such an effort is that it would bring the experience of the HPCRCs in developing their computing environments to bear directly on strategy for bringing up the initial production environment at NERSC.

An orderly transition for the majority of the NERSC user community to parallel computing will take some time. The combination of the current resources of the Access Program and the HPCRCs make a compelling basis for accomplishing this goal. A plan for implementing the transition will have as its first major milestone the effective use of the early production environment on a 100-500 GigaFlop massively parallel computer in the Access Program.

The next section is therefore dedicated to what defines an early production environment. Since the DOE/ER community will be involved heavily in building the ER arsenal of parallel applications, it is clear that the environment will have to permit the community to create, test, debug and analyze programs as well as to run the codes which are already in production status. A description of that computational infrastructure follows.

III. Implementation: Supercomputer Production Computing Environment on Massively Parallel Computers

A. Introduction

It is commonly accepted today that applications drive software development. Simply stated, this means that operating systems, libraries, software tools, indeed the entire production environment evolves according to the dictates and requirements of the users motivated to extract the most from their applications. Since our community has very strong ideas about how the production environment should look and feel, the only way that such an environment will develop is if the user community is involved. The period spanned by the transition from primarily sequential to primarily parallel computing for our user base is brightened by one extremely positive factor at NERSC: the Cray Research C90. Its presence will allow for a smoother evolution because for several years it will provide a competitive alternative for many applications. This is not a luxury; it is a necessity, given the expectations and needs of ER users. Simple prudence and good sense dictate that advantage be taken of this situation. All these considerations form criteria for "an initial production environment" to be provided alongside the familiar Cray C90 production computing facility.

Therefore, taking into account the rapid evolution that we expect to see, we will seek to provide a set of standards to be met by a system just after it has stabilized on the floor (call this the initial production environment (IPE), well within three months of installation) and to specify other requirements for somewhat later, which could be described as a general production environment (GPE). GPE capabilities are expected to evolve on the machine through the vendor's efforts in software and hardware areas, through software development at the HPCRCs and through research sponsored by the Advanced Software Technology and Algorithms (ASTA) component of the HPCC Program. These GPE capabilities should be available well before the end of the useful life of the machine. During the six months, after the machine begins operation, one could expect that the machine would be lightly but increasingly loaded, populated primarily with Grand Challenge codes developed by NERSC users on other platforms. These users will be very influential in determining the nature of the mature production environment. This period will be a critical time for NERSC staff and users to come up to speed in their understanding of the use of the initial production environment

Because massively parallel computers are a new context for distributed computing in which tasks will be divided between workstations and the massively parallel machine to increase interactivity, workstation software is another critical part of the computing environment for parallel computing. For example, distributed calculations in which the visualization is done on a workstation while the bulk of the calculation is performed on the massively parallel machine will be common. Moreover, the critical question of steering a long simulation on the massively parallel computer during its execution must be addressed. Such calculations are done today on conventional supercomputers and require that advanced visualization of intermediate results from the mainframe be performed on the workstation and that appropriate communication take place to change the course of the calculation. Limited tools for this purpose exist for conventional machines, but for parallel

machines they are almost nonexistent. Our community's requirements for the look and feel of an acceptable production environment will drive the development of such tools.

B. Initial Production Environment (IPE)

Most of the software technology required to create a functional initial production environment will be well in hand by 1994. Its development will have been driven by the vendors themselves, who have been listening to the demands of experts in advanced scientific computing at the DOE laboratories and other government institutions which traditionally procure early serial number mainframes. However there is need for research and development of tools and operating system (O/S) capabilities from sources independent of the vendors themselves in order to understand generic issues more independent of particular machine design than might be pursued by a vendor. Two of the components of the Advanced Software Technology and Algorithms (ASTA) effort will be useful in the development of production environments. The first effort is driven by the two HPCRCs themselves, which are obviously interested in creating functional environments for their primary computing hosts, the Intel Paragon and the Thinking Machines CM-5. The second area of effort is to be found in the Software Components and Tools area of ASTA. While most of the ASTA research will be felt in the more highly evolved general production environment, even from the narrow point of view of the initial production environment, ASTA research can make a significant difference in the following areas:

- *Software tools...* Research in performance, instrumentation and debugging tools would have an immediate impact on initial production environments. Development of efficient data parallel programming models for both remote access and distributed memory MP computers is another fertile area of research which will increase the efficiency of an initial production environment. The development of a common communication interface (for message passing) to facilitate portability between various MP computers would be an asset in the porting of message passing applications to the access center MPP. Workstation software for distributing calculations between workstations and the MP computer is another appropriate target for early research.
- *Systems research...* Studies of job scheduling algorithms, and an understanding of proper procedures for reliable job (and process) recovery after scheduled and unscheduled down times are critical areas which need to be understood.

(a) Computing Capabilities and Machine Attributes

Certain basic capabilities are viewed as necessary by the access community. These involve both hardware and software and are described below.

Clearly, the programming models and the underlying architecture must be sufficiently well designed so that some minimal levels of efficiency are built into the computational environment. One rough way to measure this would be to insist that for some appropriate (characteristic) set of codes a reasonable level of efficiency must be attainable using at least one of the resident programming languages on the MPP. A suggested target for performance might be 15% of peak for all or most of the codes within this set of applications. At this time, we do not seek to define the characteristic set of applications. This will be defined at the time that the procurement process

begins. It will represent the input of a broad range of users. It would be the responsibility of the competing vendors to work with the owners of the applications to write the codes in the appropriate programming language and meet the 15% objective.

A second attribute relates to the hardware alone. For any machine acquired, it must advertise a peak speeds two orders of magnitude greater than the C90 (single processor mode). This request is driven by the preceding section devoted to ER applications requirements, in other words, the problems which scientists hope to attack require this peak capability. The memory requirement is 128 MegaBytes per CPU and that too is driven by the needs of the more memory-intensive Grand Challenge problems researchers hope to attack soon.

Finally, a suite of benchmarks will be chosen (representing computational kernels) which must meet clearly defined performance criteria. The same will hold true for resident libraries. The benchmarks and the criteria to be applied will be determined at the time of the procurement and will involve ER user-base input.

(b) Programming Models

Since the user base is large and representative of all areas of ER, one can expect that (1) the variety of codes ported to the MP computer will span the spectrum of applications worldwide, and (2) a significant fraction of users are more interested in the computer as a tool rather than a computational experiment and cannot be expected to go to great efforts to extract efficiency from their codes. There are three main programming models for MP computing: message passing, data parallel, and shared or globally addressable memory. Experience has shown that all these models are useful, and in fact different phases of a particular problem can map best to different programming models. It is desirable, therefore, that the computer support all three models, and better still, if it supports a mixed programming environment. At a minimum, it is expected that the system support both data parallel and message passing programming languages. There are both hardware and software approaches to implement such support, and vendors have chosen different paths. Thus, a minimal requirement is that the vendor provide the software (and hardware) allowing the user to program in these styles. Some vendors have implemented versions of these program models which cannot be described as "popular" or "standard". Clearly, what would be required is that the programming model, even if it is vendor specific, have a syntax which maps well to a standard, popular paradigm, thereby minimizing porting problems. Beyond this basic capability, it would be preferable if the user could utilize more than one model within one program, if desired. We would expect that all these models be accessible from Fortran, and suggest that every effort be made to make them all accessible from "C" as well. We anticipate that the capability to utilize multiple programming models within one application will be a feature offered by a number of vendors in the 1994 time frame.

(c) Operating System (O/S)

The O/S would feature a UNIX programming environment. In its initial production configuration, it should provide batch processing support, configuration management support, parallel I/O, and a software interface for X-window based debuggers and performance analyzers. We will touch on

some of these below. It is not insisted that the tools run on the MP machine itself, merely that they be available upon a host which can interface with the MP computer. From our discussions with various vendors, we would be surprised if any of the capabilities described below would not be available on a number of offerings beginning in 1994.

Batch Processing...

A batch system that is similar to NQS should be in place initially. The user would, for instance, submit a job to a queue characterized by "*N*" processors and a specific time limit. If there is only one batch partition, then a scheduler must be in place to swap jobs in and out (otherwise a long running job would shut out other users). We term this swap out capability "roll-out." Refinements with such a scheduler would not be required initially.

I/O...

High speed parallel I/O functionality is required, meaning that I/O will be able to be written to multiple disk drives in parallel. This capability ought to be available initially. These rates should be high enough for the swapping of large jobs to disk to be feasible. In the competitive procurement process a minimal I/O rate should be specified that will provide for acceptable roll-out capability.

Roll out and Scheduling...

Basic job roll-out to disk capability will be available initially. The alternative to this is either to kill a long running job which is blocking other users from access to the machine, or to let it run to completion. With the capability to roll-out, provided by the vendor, it is then possible to implement a space- and time-shared gang scheduler.

Dynamic Partitioning...

Dynamic partitioning by the system is necessary. Among other things, this will facilitate the development of a scheduler which can roll out jobs, re-partition, and then roll in jobs that have been waiting to run. Jobs which have been waiting may require fewer or more nodes than the jobs which they are replacing, creating a requirement for dynamic partitioning capability. Maximum flexibility in the use of the machine argues that the machine be partitionable down to one node, since this would simplify the creation of a space- and time-shared environment under the control of a scheduler. The ability to run a job on a single CPU is considered to be critical to the access center user base, as this capability facilitates the transition from sequential to parallel applications.

Configuration Driver And Support For Machine Degradation...

A configuration driver should keep track of failed PEs to avoid crashes. Support for graceful machine degradation if nodes fail, and some dynamic mechanism for mapping them out (and perhaps dynamic substitution of

extras (spares)) should be possible. At a minimum, if a node crashes, the machine should have the capability of being brought up minus the offending node.

(d) Tools

MP computing complicates applications programming and debugging considerably. Keeping track of the activities of the various CPUs, addressing synchronization, determining efficient data distributions, and working to squeeze efficiency out of a shared loop all require software tools. The task would be next to unaddressable without these aids. Nonetheless, it would not be realistic to require that a complete suite of tools be available initially. We will describe an initial capability.

Debuggers....

By far the most critical tool is the debugger. The features which must exist initially would include an X-window-based source code debugger which is applicable for shared, data parallel and message passing code (for both Fortran and C code). It should allow single CPU job debugging. For a shared code, the user should be able to open windows on each process and look at the values of shared and private arrays without difficulty. One should be able to reassign the values of variables, and set conditional breakpoints by looking at displayed source code. It would be useful to be able to plot 1-D arrays or slices of 2-D arrays, and generate contour maps. If the code aborts during execution under the control of the debugger, provision should be made to allow a post mortem. These capabilities are being actively pursued by vendors today, and in conjunction with work being undertaken in the ASTA program, we anticipate no difficulty in finding numerous vendors offering this kind of capability.

Performance Analysis And Enhancement....

This kind of tool is not as clearly definable as the debugger, but there are many existing examples. Some vendors choose to create several tools to achieve all the possible ends; others attempt to create one environment for performance analysis and enhancement. Examples of basic functionalities that are currently being implemented by vendors follow:

- (1) The user should be able to generate an event log containing the desired performance data for the program by instrumenting the application program.
- (2) There should be an X window based graphic user interface—this would show what “happened” during a particular run—perhaps through a skyline display to show which regions of code are executing inefficiently. The user should be able to search for specific events, measure the time between events, scroll through the event log, etc.
- (3) There should be some kind of display of hardware counters which provides hard information to the user concerning cache hits, network delays, etc.

- (4) Some mechanism to relate all of this information back to source code (a code browser).

These represent basic functions to be found in performance analysis tools.

(e) Software Libraries

The software libraries should be given MP processor specific capabilities. These would include stack management, memory management, array intrinsics and I/O support. These capabilities should be well-evolved by 1994. In fact, many libraries and tools are available already. For example, the popular visualization tool AVS is marketed and supported for a variety of massively parallel machines.

Math libraries are of two types, sequential and parallel. It is expected that a full complement of sequential (processor local) math libraries be available to the user. These include the complement of libraries that the users are accustomed to on the Crays. Examples include the NAG, SLATEC, IMSL etc., and a number of these are already available on some parallel platforms. All the desired parallel libraries may not be available at this point, and vendors are willing to work with users to develop sought after parallel mathematical routines. Some third-party application codes, such as GAUSSIAN or NASTRAN, may not be available initially. These codes are typically leased from software vendors who may not provide parallel versions of their codes until programming model standards exist, but demand for them is increasing.

(f) Disks and Tertiary Storage

Adequate swap space is required to facilitate time sharing. We recommend at least 900 GigaBytes disk storage. Data bandwidth to disk was discussed earlier. Adequate tertiary storage capability is as critical as rapid I/O and adequate disk space. A review of the appendices indicates that many simulations, from the Numerical Tokamak to Climate, will require huge storage capabilities. For instance, a moderately resolved climate model simulation (requiring a 240 hour run time at a sustained 100 GigaFlops) would generate about 23MB per second for a grand total of 20 TeraBytes of data. Clearly, this information has to move from the MPP to disk and then from disk to mass storage in some sustainable manner during the run, or the calculation is not feasible. Once the data is stored, it has to be accessible to be useful. This means that, for instance, data should be able to be selectively and rapidly removed from the storage device and fed to a graphics engine to generate movies and graphics stills. It is not currently feasible for 20 TeraBytes to be allowed to remain on a practical archival device indefinitely. However, it should be able to exist on the storage device long enough to allow the data's assimilation by a graphics engine or some other post processor. We recommend a minimum of 40 TeraBytes tertiary storage.

A 20 TeraByte data set would take up a significant fraction of any archival storage facility which could be practically deployed with today's technology. If we use massively parallel computers as we have used conventional supercomputers, the larger memories and higher speeds of the new machines imply that we will have to deal routinely with such large data sets. One of the changes which will accompany the transition to massively parallel computing will be a change in the style

of the use of the new supercomputers. We will almost certainly see a shift to *recomputing* data rather than storing it. The trade-offs between those two strategies will change as the ratio between the time to retrieve data and the time to compute it changes.

We cannot, however, ask for a change in the style of the use of mass storage to extricate us from the dilemma of data storage. As the massively parallel computing arena matures, new types of data storage devices, such as optical tape, are appearing. With a single-reel capacity of a TeraByte but a slow writing speed, these devices could form the last and largest tier of an extended storage hierarchy. Such hierarchies and the software to run them are part of DOE's current research efforts in high performance computing under the ASTA program.

The 40 TeraBytes of archival storage recommended here should be considered a necessary interim solution to the data management problem in the massively parallel computing environment. A strategy for longer term solutions is the subject of current study.

C. General Production Environment

The general production environment will evolve out of the efforts of the vendor, ASTA researchers, and the access center staff. In addition to the research areas described in the previous section in which ASTA would have impact, we believe that studies in the following areas will have a positive influence in the general production environment. We provide only a few examples:

- *Software tools...* Research into higher level languages and advanced compiler technology is critical. For instance, any data parallel model implemented should have a data distribution tool associated with it which would suggest distributions and conduct efficiency analyses. Clearly, work should be conducted in the area visualization and the management of the data which will be generated by the MP production environment.
- *Software libraries...* An area of greatest import is the development of parallel linear algebra algorithms. This would facilitate the implementation of implicit differencing techniques in user applications and widen the range of problems accessible to MP processors.

(a) Operating System

Batch Processing...

The batch system will be fully integrated with a scheduler to allow a time-shared batch processing capability.

I/O...

Programming models and I/O system will allow the applications to write into a parallel file system

Checkpoint Restart and Interactivity

Checkpoint and restart capability—a more complex problem than simple roll-out—would be expected to evolve and be available. The user must have the possibility of running with “*N*” nodes, checkpointing, and then restarting with “*M*” nodes. A fully interactive time shared environment would be expected to be in force, under the control of a mature partition scheduler. A number of partitions consisting of only one processing element would be available for users interested in using one node only, and the scheduler would assure that many (perhaps hundreds) of users could time share simultaneously.

Loader...

It should be possible to set the number of PEs at load time, or even at execution time, which is by far the best option.

(b) Tools

Both the debugger and the performance analysis suite should reflect a mature operating environment. This should emphasize features which accept that users are running applications on many

processors, and help them to understand their codes performance characteristics. Some possible features follow:

Debuggers

A color debugger in which elements of arrays map to colored pixels, so the user gets a colored map describing his array, would be useful. Since the MP computers will run problems in the gigaword neighborhood, some arrays will be very large indeed, and it will not be possible to look at all the numbers in shared arrays. There should be some mechanism for looking at private objects. In a problem using 1000 nodes, a user may wish to look at the private variable "x" in each node. A graphical system for looking at all of the x's should be possible. In a 1000-node run, it is impossible to open a window on each process. Typically, a user will not be debugging an application with 1000 nodes—this would be very inefficient (given projected swap rates). However, occasionally, debugging with a high number of nodes will be necessary, because smaller runs will not reveal problems, and it is in this operational mode that the features described above can save the day.

Performance Analysis and Enhancement

Some possible functionalities follow:

- (1) An automatic code transformer, which allows the user to select and apply code transformations (for instance reorderings of DO-loops). The code could then be re-run using the performance analysis tool to see the efficiency gained or lost.
- (2) A static performance analyzer, which looks at code (subroutines) and suggests changes or provides a static performance analysis based on known machine performance characteristics. The user could conceivably ask questions concerning how efficiency would change if the number of processors were doubled, or if an array were to be distributed differently.
- (3) A data distribution tool which would suggest a particular distribution; the user would then apply either the static or dynamic analysis tool to determine the efficiency of resulting code.

Workstation interface tools

Distributed visualization tools for viewing large data structures or the results of a calculation in progress on the MP machine will be an important part of the general production environment. The standards for such

software interfaces are almost completely absent currently for MP machines, but there are powerful models in the conventional supercomputer environment. UNICHEM, a code which is distributed between high-end graphics workstations and mainframes, is an example. Developing the underlying interface software on which such codes are written involves work both on the workstation side and on the MP processor.

(c) Software Libraries

Parallel mathematical libraries are the Holy Grail of parallel computing. General libraries which run on many parallel machines are not available and will not be available until standard programming languages coalesce, allowing for code portability. Even so, the problem of tuning for a particular architecture will remain. Encouragement has to be provided to vendors to create parallel linear algebra libraries for their products, but to expect a suite of routines to be available even by 1995 would be completely unrealistic. It is possible we will see a parallel version of the BLAS by 1995; however, it is not likely that a complete set of sparse matrix and generalized eigenvalue solvers would be available for some time after that. This is an area where users must work with the vendor to drive the software development and in which the access center user base could be particularly effective. (An examination of the requirements appendices for plasma physics suggests that the speed up obtained from MP computers utilizing explicit methods could not compete with a scenario employing implicit techniques and matrix solves). Clearly, work by ASTA researchers, if successful, would have great impact in this area. This points to an aggressive program by the vendor to work with users to isolate the most common linear algebra problems and to develop a library utilizing ASTA developed technology.

IV. Implementation: User Transition to Massively Parallel Computing

A. Introduction

This section focuses on transitional issues, both for the users and for the access center. Any strategy to be developed must depend on certain assumptions concerning the time that will be available to effect the transition. The assumption that we make here is that for the Access Center, the transition period is until roughly the middle of 1995. However, some individual users could take the entire lifetime of the C90 to make the transition. It will probably require six months for the initial production environment outlined in Section III to evolve into a general production environment. Bringing the staff up to speed, preparing course documentation and examples, and developing a staff educated in the use of parallel computers in the areas of operating systems, applications and tools, should be complete well within a one-year time frame.

Summary of Essential Elements in the Transition

1. Agreements with vendors to expedite the education of access center staff in the MP environment and to push the production environment along the lines desired by access center users.
2. Interaction with users through applications collaborations, courses, workshops, and an ERSUG core user group that maintains contact on a day-to-day basis with access center staff. This group of users would report to the Executive Committee of the Energy Research Supercomputer Users' Group (EXERSUG) on a regular basis.
3. User Services area will be augmented by a group dedicated to MP computing issues, from converting applications to developing detailed understanding of tools. User Services group will develop courses and documents in the new areas, as well as provide a database of codes which the users will have available as learning tools. The consulting group will take pertinent courses and work with vendors to learn about the MP environment and develop teaching programs for users.
4. Collaborations with HPCRCs (and possibly other government Labs). HPCRCs can be of great use to the access center in developing its infrastructure. They would be particularly useful in the areas of (a) educational support for staff and users, (b) sharing and development of software with access center, (c) allocation of modest computer resources to NERSC to allow access by some access center users, (d) advice on the configuration of access center hardware and software. These areas will be amplified upon in the relevant sections that follow.

B. Transitional Issues Facing The Access Center

(a) Machine Operation

- Agreement with vendor to develop procedures for passing to access center staff relevant information concerning operation of the machine.
- Availability of vendor experts in machine operation and O/S on site.
- Recruitment of staff who have experience in a MP environment.
- Development and maintenance of strong collaborations with the HPCRCs. Their experience in operating large machines will be essential to access center success. The possibility of exchange programs should be considered. In particular, it would be useful if access center staff could spend some time working in the machine room of an HPCRC to gain experience in tasks associated with machine booting and hardware configuration, etc. The value of this kind of hands-on experience is hard to quantify, but should manifest itself in enhanced machine reliability at the access center. Since the HPCRC would be highly experienced in these areas, this kind of exchange would be useful in the months before the arrival of a machine at the access center.

(b) Production Environment

Vendor is responsible for delivering an initial production environment as specified in Section III. There are three avenues to follow to facilitate the development of a general production environment: (1) collaboration with the vendor coupled with (2) close working relationship with the users as they port their applications to the MPP and (3) collaborations with other laboratories.

- Access Center Staff and small selected user base will assess this environment and work with vendor to seek changes as necessary. The object is to have functionality acceptable to average users within six months. This would represent the initial production environment.
- Access Center Staff must understand tools and machine capabilities. This could be achieved by the creation of "toolmasters." One way to create a toolmaster is to have staff work directly with vendors in their tool development by providing input on user requirements and by trying out early version of the software.
- Mechanisms might be found for staff to work directly with vendor on software for O/S upgrades. The purpose is to develop expertise in the O/S at the Center, not to get into the business of creating software.
- Staff working in applications areas (with users) will provide a source of high level information to vendor and access center users on functionality of production environment.

- Just as the HPCRCs can provide valuable experience in the area of machine operation (above), it can be useful in providing experience in creating, maintaining and developing a production environment. A close working relationship with at least one HPCRC should not be difficult to develop. The following areas of intercourse and collaboration should be explored: (a) The HPCRCs will have developed through experimentation a fundamental understanding of what works well in the global environment, including the configuration of disk farms, graphics post processors, storage, as well as the MP processor itself. It would be useful for access staff to make extended visits to HPCRC sites to understand this infrastructural environment. (b) HPCRCs have provided extended classes and workshops (for example the summer workshop at ACL). It would be sensible if access center staff would take advantage of these programs. (c) Staff at the HPCRCs have been involved in the development of software tools and programming models for parallel computers. This has also been an area of active work at NERSC (due to its MPCI experience). Clearly, there is room for coordination of efforts in the development of critical software such as a gang scheduler. In the area of programming models, software interfaces should be created which will allow the simple porting of applications across machines. This would make particular sense for applications using message passing. (d) The HPCRCs might allocate small amounts of computer time to allow access center staff to come up to speed on the programming environment of the HPCRC. In this way the center staff would become familiar with tools and local programming models.
- Through the mechanism of ERSUG. A designated group of users could have close contact with a NERSC group dedicated to MP issues. This could result in detailed reports at ERSUG on progress.
- Collaboration with DOE and other government laboratories that have purchased from the same vendor. This would be particularly attractive in the case of LANL and LLNL (sister institutions with a similar computational culture).

C. Transitional Issues Facing the Users

(a) User Services (Classes and Education)

- Documents which will introduce the users to the spectrum of tools, the programming paradigms, and the range of the "minimal production environment" must be available before the arrival of a machine. This would provide the users both with a sense of the potential of the new machine and its current limitations. This material should include motivational material, including examples of sequential codes that, through parallelization, have experienced significant speed up. A database of sequential and parallelized codes with which the users can experiment is essential. The users could then assess the work that goes into parallelization, the speed-up resulting from parallelization, and then utilize the appropriate tools (such as debuggers and performance analysis tools) with the test suite of codes. This process will give users a "feel" for the new environment as well as a basic understanding of programming models.
- The access center should provide either in-house workshops or off-site classes in use of the resource and development of parallel applications. In general, these would be overview oriented and "get started" classes, which could be put on videotape. The basic or first course material would emphasize (1) architectures, (2) programming models, and (3) tools, and would be conducted in a laboratory environment where the members would have access to X-terminals. A more advanced class would emphasize techniques for code parallelization. At yet a higher level, the Center could sponsor workshops targeted at parallelization of codes within a given discipline. For instance, in plasma physics, the workshop would focus on particle-in-cell and in fluid modeling parallelization techniques. These would be headed by scientists experienced in these areas of computational physics.
- Access center will examine the possibility of providing courses through video-teleconferencing.

(b) User Services (Interaction with Users)

We identify the following areas:

Day to Day Consulting...

In this radically new environment, the need for consulting services is clearly greater and more critical than it has been in the past. Since the user base will be exposed to a new mode of computing, the user services area will be under considerable pressure to respond to numerous basic questions in all areas, including programming models, debugging, performance analysis, libraries, etc. Users will have to understand the limited functionality of the initial production environment and at the same time be encouraged to understand it and appreciate its virtues. This will in all likelihood require a small increase in FTE levels in the day-to-day consulting area.

Collaborative Efforts With Users...

The users will frequently need help in the optimization or parallelization of codes. The kind of help required here is far more intensive and detailed than what was outlined under day-to-day consulting, would represent a new service at the access center, and should be provided to the extent that the resources are available. Two types of NERSC involvement are envisioned:

- (1) Toolmaster support. These are NERSC staff who have had the opportunity to work with the vendor and obtain a detailed understanding of the hardware and the software tools supplied by the vendor. These persons would be expected to work closely with users to tune or debug a parallel code which is giving the user problems. This kind of effort could evolve into a collaboration, particularly, if the interaction is in the area of performance analysis.
- (2) The second area would involve a collaboration in which the NERSC staff member actually creates parallel versions of user applications with the user. A number of NERSC staff would be trained in parallel programming techniques and would have a background in computational physics. This combination of skills would result in relatively rapid parallelization of the more important applications and coherent feedback to the vendor concerning shortcomings in hardware and software.

(c) Code Migration

- During the period before the arrival of an MP device, it makes sense to introduce as many users as possible to parallel computing, so as to have codes available immediately. One mechanism is for users to develop parallel versions of their codes on their own, at one of the HPCRCs, or at another site, including LLNL's TC2000 available to NERSC users. Another possibility is to make arrangements with the HPCRCs in which some computer time is made available to access center users to begin the parallelization process. Computer time would be for development, not production, to minimize impact on the HPCRC. Some of these users could develop collaborations with access center staff dedicated to work with users in the development of parallel applications. In some cases of interest, the staff could take over the complete burden of parallelization, with the user acting as a mentor for the physics and the staff member as a mentor for the parallelization.
- Some Grand Challenge users will already have parallel versions of their codes developed at the HPCRCs and at other sites. These codes will be written utilizing a variety of programming models. This is one reason why it is essential that the access center machine support all major programming models. Then it is merely a matter of translation from the syntax used on the foreign device to the local syntax, rather than a complete rewrite of the code. This process could be expedited by (1) an access center project

dedicated to working with users on their applications conversion and by (2) the consulting staff who will be well versed in the local programming languages to aid users working alone.

- Some users will have serial versions of their codes, with little or no experience in parallel computing. After attending the classes or reading the introductory documents prepared by the access center staff, the user should have the option of contacting the access center group dedicated to parallel applications. This could be as short as a few hour consultation, or a lengthy collaboration in which the user and the staff member together plan and develop a parallel code. (Clearly, the number of full-scale collaborations is a function of the number of trained, available staff, and would never be large).

V. Recommendations

This study recommends the procurement of a massively parallel computer in the Access Program capable of 200-300 billion operations per second for delivery in third quarter FY94. The only path to satisfying the needs of Energy Research for high performance computing is the use of large massively parallel computers together with the currently evolving distributed computing environment at ER sites.

The initial configuration of the massively parallel computer should consist of a machine capable of approximately 200-300 GigaFlops in order to satisfy the first level of mission requirements outlined in this document. Reasonable estimates based on current trends in technology place the availability of such a machine some time in calendar 1994. Some vendors are advertising early machines of this capability in 1993. Future requirements for computational capacity may determine that more than one massively parallel machine of this class will be needed in the Access Program.

An appropriate time table for procurement of the machine recommended here is to finalize the requirements for the procurement in the first quarter of calendar 1993, and begin the procurement immediately thereafter. Evaluation of vendors' offerings should be completed by the end of the third quarter of calendar 1993, and the machine and its configuration chosen by that time. Budget authority is required for a third quarter FY94 installation.

This study recommends that the procurement address increased data storage needs. A particularly critical part of the configuration of a production massively parallel computing facility is the integration of the resource into a sufficient mass storage facility with adequate speed of data access. Because of large memory sizes on massively parallel machines, a larger fraction of procurement should be devoted to disks than in conventional supercomputer procurements. A factor of five to ten times the primary memory is suggested as the minimum. Furthermore, the size of tertiary storage and speed of access to it must both increase to maintain acceptable response time for increasingly large files.

An important recommendation concerns the involvement of the user community in the choice of hardware. The procurement evaluation committee should consist of NERSC staff, members of the executive committee of the energy research users group (EXERSUG) and other users. In particular, EXERSUG will be consulted for recommendations for technically expert participants in the evaluation and selection process and for benchmarks to use in the evaluation. A collection of recommendations regarding the implementation of the procurement of a massively parallel computer and its associated hardware and software were also suggested in the panel discussion on which this document is based: The procurement should be competitive, and benchmarking with selected applications obtained from the High Performance Computing Research Centers should be considered for inclusion in the evaluation for an award, as should a preshipment performance test of the hardware. A procurement strategy which accounts for technology injection should be pursued where possible.

This study further recommends that the entire set of infrastructural components, such as consulting, networking, accounting, systems programming, applications program libraries, etc., be provided. These services are integral parts of a production computing environment

This study recommends that research and development for software libraries, programming tools, operating system features, etc. for a parallel computing production environment be undertaken with clear near-term and long-term goals. Because the software for production computing on parallel computers is still evolving, specific software sufficient for the period of "initial use" by a general community of users should be provided by the vendor and the computer center fielding the machine. That set of software must be improved into a fully functional, general production computing environment as quickly as possible. This entire effort can be naturally incorporated into the Software Components and Tools area of the Advanced Software Technology and Algorithms (ASTA) program of the High Performance and Computing Initiative.

A further recommendation is that once the vendor is chosen, the Access Program should work intensively with the HPCRCs to move as much as possible of the software environment they have developed to the coming production platform. An important part of the establishment of the production massively parallel computing facility is to leverage the experience and software base of the High Performance Computing Research Centers (HPCRCs) at Los Alamos and Oak Ridge National Laboratories. Their experiences, the software tools they have developed, and the user application codes which have been developed in those centers will form the basis for the transition for the larger community of Energy Research scientists and engineers to this computational environment. Particular emphasis should be placed on solving the problems facing users as they make the transition to massively parallel computing for the majority of their large-scale production computing. Consulting, collaborative efforts with users to migrate their codes, the provision for training, and the creation of transparent documentation on the use of software tools must all be provided.

This study recommends that a small percentage of the resources of the HPCRCs be set aside to help in the conversion of applications codes to massively parallel architectures. For the larger community of users of the Access Program the transition to massively parallel machines is impeded by a combination of lack of access and expert help. Prior to the arrival of the massively parallel machine recommended here, the Access Program should provide aid via consultation and programming assistance to a subset of its users who are granted access to the HPCRCs or other DOE parallel computing facilities.

A measure of the confidence placed in massively parallel computing as the future path of scientific computing is that, in the discussions which lead to this report, a scenario was discussed in which an "integrated massively parallel computing environment" would evolve. In such an environment a number of DOE Energy Research sites would have smaller versions of a supercomputer-scale massively parallel machine. If the same vendor, or at least similar architecture, were chosen for those smaller machines, it would be possible to debug and do smaller calculations on them at each site. The load on the large machine could be shifted in the direction of high capability production computing, while the capacity of the smaller machines would be used for a large fraction of smaller jobs. A key component of such a scenario, if it is to work, would have to be remote access to the large storage facility, which must accompany a massively parallel supercomputer at a central site. The satellite systems would have to be

integrated in terms of software, networking and remote archival and intermediate storage if the integrated environment is to meet the user's needs. The technologies to accomplish the integration are either present now or clearly emerging at this time. The serious consideration given to this scenario is an indication of the degree to which the user community and other experts believe that production massively parallel computing has become a reality.

The transition period from primarily conventional vector supercomputing to primarily parallel computing for the DOE Energy Research community will require the presence of robust conventional supercomputers such as the Cray C90 at NERSC. Commercial application codes (such as NASTRAN for structural mechanics and GAUSSIAN for quantum chemistry), which are important to many Energy Research scientists, will have to be available and performing well on parallel machines before those users can be asked to make the transition. For this reason, DOE's present conventional supercomputer resources will need to be maintained for a number of years, probably beyond the point when most users will be using parallel machines.

Appendix A—Computational Fluid Dynamics

Computational Fluid Dynamics (CFD) stands to be one of the primary beneficiaries of the push to massive parallelism. The availability of the computational capacities discussed above, combined with the advances in numerical algorithm technology achieved over the last fifteen years, will enable us to analyze and understand fluid flows for problems of substantial scientific and technological importance. In power plants and automobile engines, the efficiency with which the burning of hydrocarbon fuels produces useful energy, as well as the rate at which pollutants are produced, is strongly dependent on the detailed interaction between the burning and the fluid-dynamic mixing of the fuel, the air, and the combustion products. In the design of computer components, fluid dynamics drives the cooling of circuits, and is a controlling factor in the design of high-speed disk drives. The efficiency of materials processing, from the manufacture of composites to the growing of single-crystal silicon for semiconductors, is determined by a delicate balance between mixing, heat transfer, and the chemical reactions which cause the material to form. In geology, the percolation of fluids underground determine the flow of contaminants to the water table and the flow of oil from wells.

All of these applications are 3-D and time-dependent, have complex physical processes coupled to the fluid dynamics, and have complex physical geometries. On the present generation of conventional supercomputers, we are unable to compute problems of this complexity. A typical state-of-the-art CFD code running on a conventional supercomputer can produce accurate solutions to one of two classes of problems: either very simple physics models in complex 3-D geometries, or complex physics models in simple, primarily 2-D geometries. An example of the former are codes such as Boeing's TRANAIR, which can solve equations for a very restricted class of fluid flows—steady inviscid compressible potential flow—with no additional physics, but in arbitrarily complex geometries, such as the exterior of a complete aircraft. Examples of the latter are a variety of commercial and national laboratory engineering fluids codes which support a variety of time-dependent fluid dynamics and other associated physical models for processes such as chemistry, heat transfer, and phase change, but which can produce results with engineering accuracy for only 2-D or extremely simplified 3-D geometries.

A massively parallel computer, delivering 200-600 GigaFlops, has the potential for taking us from our current capabilities to being able to solve the problems listed above. Essentially, this is a move from two to three-space dimensions. On the face of it, the two orders of magnitude increase in performance represented by this advance would not be sufficient. A fully-resolved 2D calculation performed on a uniform finite-difference grid typically requires a 1000 x 1000 grid in two dimensions, and can be done on the current generation of supercomputers. Thus, a comparably-resolved 3-D calculation would require at least a 1000 x 1000 x 1000 grid, a factor of 1000 more computation and beyond the factor of 100 to be obtained from a massively parallel machine to be delivered before 1995. A second question that must be addressed is to what extent that the numerical methods under consideration can be mapped onto a massively parallel architecture.

The answers to both of these questions lie in the detailed mathematical properties of the problems being solved. The first of these is locality: disjoint regions in space are nearly decoupled. Locality is a general property of all of the equations of fluid dynamics, inasmuch as they are described by

partial differential equations of the three classical types: elliptic, parabolic and hyperbolic. In hyperbolic equations, finite propagation speed defines the appropriate notion of locality. The effect of large gradients in a localized region in space at a given time can influence at some later time only those points that can be reached by the waves propagating at finite speed from the original singularity. The form of locality appropriate for elliptic and parabolic equations is that of local regularity, i.e., that the smoothness of solution to an elliptic or parabolic equation in some region is determined only by the smoothness of the right-hand side in that region. For example, in solving Poisson's equation for the field induced by a localized charge distribution, the solution is harmonic in regions disjoint from the charge distribution; in particular, it is infinitely smooth, with rapidly decaying derivatives. More generally, the coupling between disjoint regions containing large solution gradients for elliptic and parabolic is mediated by such smooth functions.

The second property of the equations is intermittency: that typical solutions consist of small regions where the solution gradients are large, embedded in regions of smooth flow. Typically, intermittency is observed in situations where the dissipation terms are small, relative to the inviscid dynamics. In that case, the large gradients are initially generated in small regions by the non-linear terms, or by the dissipative terms themselves, acting as a singular perturbation. Examples of the former are the formation of shocks in initially smooth flow; of the latter, the narrow shear layer formed by the separation of a viscous boundary layer around a blunt body. The fact that the regions of large gradients remain confined to small subsets is a consequence of the presence of small dissipation terms as the dominant form of regularization in fluid dynamics. In regions where the solution becomes nearly singular, the effect of the dissipative terms become large, leading to decay of the singularity. This is in contrast to what occurs with dispersive regularization, as is the case for many problems in plasma physics, in which the near-blowup of gradients due to focusing is converted into a broad band of high frequency waves. The size of the regions occupied by large gradients in fluid dynamics is quite small; shocks and other fronts locally look like surfaces (possibly thickened), whereas large gradients in the vorticity are thought to occupy a vanishingly small volume as the viscosity goes to zero.

Modern algorithms for CFD exploit the locality and the intermittent nature of the large gradients to maximize the overall accuracy per unit of computational cost. Intermittency means that large gradients, which require the largest density of grid points or other computational degrees of freedom, are concentrated in relatively small subsets of the total computational domain. This problem has led the designers of CFD algorithms in recent years to consider adaptive algorithms, i.e., algorithms that concentrate computational degrees of freedom in the regions containing large gradients and reduce the number of computational degrees of freedom where the gradients are small. In adaptive mesh refinement, for example, mesh points in a finite difference calculation are inserted and deleted dynamically as the calculation proceeds in response to the changing conditions in the flow. In two dimensions, the use of such algorithms have been shown to reduce the cost of calculations by a factor of 10 or more; preliminary results indicate that this will continue to hold in three space dimensions. Thus, we expect that the 100-fold increase in computing capacity represented by these massively parallel machines, combined with the use of adaptive algorithms, will enable us to actually obtain breakthrough-level results for the problems described above.

In a similar fashion, the locality properties of the equations of fluid dynamics allow us to map these calculations very effectively onto massively parallel architectures. Locality leads to

numerical algorithms in which the bulk of the computational effort is used to compute the dependence of the solution at a point on nearby points in space. In explicit methods for time-dependent problems, as are used to solve hyperbolic equations, the entire calculation is localized. In the case of implicit methods for elliptic and parabolic problems, local regularity leads to iterative methods such as multigrid for which the bulk of the computational work is local, with only a small fraction of the computational degrees of freedom being used to represent the (infinitely smooth) nonlocal coupling. Thus, a natural strategy for parallelizing CFD calculations having these properties is to distribute spatially disjoint regions onto different processors. This leads to computations in which the ratio of time spent doing floating point computation to that in performing synchronization or communications is large, leading to effective use of the parallel architecture. This type of computation can be done in a data-parallel fashion, with individual gridpoints being distributed to different processors, or in an asynchronous MIMD fashion, with patches of gridpoints being distributed to different processors.

Thus, the presence of a massively parallel computer would allow CFD to make a substantial impact on a variety of problems of substantial importance to DOE. However, the way in which it will make this impact is by allowing the development of a whole new generation of simulation codes with qualitatively different capabilities, rather than simply scaling up existing codes by adding more grid points. Thus, for a massively parallel machine to be useful in achieving this goal, it must constitute an appropriate environment for developing such advanced software. It must provide appropriate languages and tools for expressing the programmer's intent regarding parallel execution and data layout that are sufficiently general to support the kinds of dynamically-changing data structures represented by the adaptive methods discussed here. It must also provide tools such as object-oriented programming languages to manage the software complexity of modern CFD algorithms. These calculations will generate enormous amounts of data (up to 16 GigaBytes per time step, or 10^{13} words per run). We must be able to interactively browse this data to extract useful information from the calculations. Finally, the implicit assumption is that we will be able to achieve a substantial fraction of the peak performance of the machine—initially, at least 15%—to achieve the goals we have set out here.

Appendix B—Lattice Gauge Theory

Lattice gauge theory is a subfield of theoretical particle physics which uses supercomputers to study quantum field theories. With only a slight over-generalization, one might identify two classes of problems which use the supercomputer. The first comprise relatively small-scale investigations of points of principle, perhaps to provide qualitative understanding of a particular theory, or to see if it is feasible to compute some quantity, or to study a new algorithm. The second class of problems comprise “production runs” aimed at the reliable extraction of quantitative predictions which may be of relevance to experiment. These are almost exclusively performed in the context of lattice quantum chromodynamics (QCD), the lattice field theory believed to describe the interactions of quarks and gluons. A competitive calculation usually requires several thousand Computer Recharge Units (CRUs). Typical outputs include: the masses of the proton and other hadrons, studies of hadronic structure, matrix elements of weak interaction operators, and thermodynamic parameters of the transition to the quark-gluon plasma. For several of these quantities, notably weak interaction matrix elements and the chiral phase transition temperature, lattice QCD calculations are accurate enough to play a significant role in the confrontation of theory and experiment.

There are five parameters which govern the quality of the results of lattice QCD runs: (1) the number of points in the 4-D grid, (2) the resolution of the grid, i.e., the physical spacing between adjacent lattice sites, (3) the quark masses, (4) the statistics of the Monte Carlo sample, and (5) whether or not one includes the effects of vacuum polarization. No simulation at present—or in the next few years—can work in a regime where all five control parameters are in their physical or asymptotic regime. Thus some extrapolation is always required to extract a physical answer. Of these five parameters, the one that costs the most to control is the inclusion of vacuum polarization. Simulations which make the uncontrolled approximation of ignoring this effect (quenched simulations) require about two orders of magnitude less computer time than those in the correct theory (dynamical simulations). Current (1992) state-of-the-art calculations are performed on 4-D grids of size up to 32^4 with typical lattice spacings of about 0.05 Fermi. Those which use the quenched approximation can be run on traditional vector supercomputers; those using the full theory with dynamical fermions are currently run almost exclusively on massively parallel machines.

Like many other physical problems, lattice QCD is very well suited to parallel machine architectures. The key computational kernel is the solution of large sparse matrix equations with random coefficients. These are solved by iterative techniques such as conjugate gradients or minimal residual, possibly with some preconditioning. Since the structure of the matrix is regular, this problem is ideal for the data-parallel model of computing. Furthermore, since the fundamental nonzero blocks are three-by-three complex matrices, each piece of data which is communicated from one site to the next participates in six multiply-add operations. Thus, lattice QCD can usually tolerate a lower communicate-to-compute ratio than some other problems. Finally, the overall size of the programs is not large, typically a few thousand lines. Therefore, it is usually feasible to handcode the inner loops to optimize performance of both computation and commutation.

For these reasons, several lattice QCD groups have found it worthwhile to invest the time in building and programming their own custom designed parallel machines. Until 1992, these home

built machines were the dominant computing resources on the international horizon. Examples include the 256 node machine at Columbia University, the GF-11 at IBM Yorktown Heights, the ACPMAPS machine at Fermi National Accelerator Laboratory, the APE at the University of Rome, and QCDPAX in Japan. Each of these machines is capable of sustaining in excess of a GigaFlop (more than 6 GigaFlops in the first two cases), and are primarily or exclusively dedicated to lattice QCD. In the current time frame, commercial machines have just caught up to this level of performance, and there are lattice QCD groups with exclusive or dominant use of CM-2s from Thinking Machines, as well as i-860 based machines such as Intel's Delta and Meiko's Computing Surface. Performance on these machines is competitive with the custom-built machines, e.g., lattice QCD can sustain about 6 GigaFlops on a full 64K node CM-2.

Given this migration to parallel machines, what is the current role for traditional vector supercomputers in lattice QCD? The strength of these machines has been their relative familiarity, ease of use, and ease of access. There have been examples (in Germany and Japan) of runs corresponding to a few months of dedicated Y-MP-8 or HITAC-S820/80 time, but to a large extent, such vector machines are becoming servers for small-scale tests of ideas and algorithms. In this role they provide resources to researchers who either have no access to a large parallel machine, or do not want to tolerate the environment of a parallel machine, or simply do not require a large amount of horsepower to perform their computation. In the last category are both tests of concepts and algorithms, as well as a few hybrid calculations where the computationally intensive portions have been performed on a parallel machine, while the vector machine performs a post-analysis requiring only hundreds or thousands of CRUs. This post-analysis may be most suitable for the conventional environment either because it is easier to program on a vector machine, or because of the availability of certain software packages, or because of easy access to mass storage.

There is every indication that there will be dedicated lattice QCD engines in the 100 GigaFlop range in the 1993-1995 time frame. In Italy, the APE-100 project is funded and under way, and aims to be running on a 100 GigaFlop custom built machine. In Japan there are plans for a 500 GigaFlop follow-on to QCDPAX. In the U.S.A there is an even more ambitious effort being put forward, which proposes to build a parallel machine capable of sustained speeds in excess of a TeraFlop on lattice QCD code. This would be constructed in a collaboration with Thinking Machines, MIT, and members of the QCD TeraFlop Collaboration. Chip design is already under way, though approval and full funding of the project is far from certain. Nevertheless, it is clear that resources of this magnitude will soon become necessary to stay competitive on the international scene.

As discussed above, to get a physical answer from a lattice QCD calculation, one must at present extrapolate in three parameters: lattice spacing, lattice volume, and quark mass. In addition, although one may use the "quenched approximation" in a first round of calculations, in the end one should repeat the calculation in the full, expensive theory with dynamical quarks. Each of these effects leads to a systematic error, and even a TeraFlop-year of computing would not be enough to work in the physical limit of this parameter space. Therefore, one must decide how to spend limited computing resources, depending on which of these sources of error is the most important. This decision will depend on the particular physical quantity one wants to compute.

The prototypical example of a lattice QCD calculation is the hadron spectrum, i.e., the masses of the proton, rho, pion and other related particles. In this case, one of course knows the answers nature has given. Thus, a calculation gives a test of the QCD theory, and of lattice QCD in particular. It is probable that no one seriously doubts that this test will be passed, but doing so will give added confidence in other lattice QCD calculations where the answer is not known.

Figure B1 shows some typical recent data concerning the hadron spectrum. The plot shows proton mass versus pion mass, where both are measured in units of ρ meson mass. In the unphysical limit where all quarks become infinitely heavy, we get the point at the upper right of the plot. As the quark mass decreases to the physical point where $m_\pi/m_\rho = 0.18$, one hopes the data will converge towards the physical value of the nucleon mass, marked in the lower left corner. Plainly it is plausible that lattice QCD will correctly reproduce the correct value of the proton mass, but to check we must decrease the quark mass until $m_\pi/m_\rho = 0.18$.

To avoid severe finite volume effects, as one decreases quark mass, one must at the same time increase the number of points in the lattice. One can estimate that to perform the definitive calculation of the hadron spectrum one will have to go from current lattice size (about $33^3 \times 64$) to sizes of about $128^3 \times 256$. Counting flops and bytes, the resources needed for such a calculation are about 300 GigaBytes of main memory, and about 1000 TeraFlop-hours of runtime. Such a definitive calculation would be feasible in the 1995 scenario discussed here. In the interim, one might imagine a calculation on a $64^3 \times 128$ lattice requiring only 30 GigaBytes of memory and 30 TeraFlop-hours. This latter calculation would be feasible in the 1993 scenario.

The spectrum is a slightly artificial example of a computable quantity, since we already know the answer. There are many other physically interesting quantity worth computing in lattice QCD, e.g., electromagnetic form factors, weak matrix elements, spin and flavor structure, heavy-light systems, and the spectrum of exotic mesons. The demand on CPU and memory for each of these is of the same order of magnitude. Therefore we will pick only one example to highlight here.

As an example of an quantity whose determination would have major impact on the rest of the field of particle physics, in Figure B2 we show recent results for a particular weak interaction matrix element known as B_K . Here the dominant source of systematic error is the extrapolation to zero lattice spacing. The plot shows two possible extrapolations, linear and quadratic. To improve the situation, one would again like to run on a lattice of size $128^3 \times 256$, but with the physical lattice spacing decreased from its current typical values of about 0.05 to 0.1 Fermi down to 0.025 Fermi. The memory requirements are essentially the same as for the spectrum above, namely 300 GigaBytes, while the computer time requirements would be about twice as much, or 2000 TeraFlop hours. Again, a very useful intermediate step would be to go halfway in the parameter space, working on a $64^3 \times 128$ lattice which would require about 30 GigaBytes and 60 TeraFlop hours.

One of the strengths of the NERSC environment is the easy access to mass storage. This is of particular relevance for lattice QCD calculations, because the basic building blocks of different calculations may be essentially the same. Thus, the intermediate data which is generated in a spectrum or matrix element calculation can be saved and "recycled" for other purposes without a large amount of new demand on resources. At NERSC there are already several examples of such reuse of data, often involving a transfer between different collaborations of physicists. In some

cases data has been generated and then reanalyzed at NERSC; in others, data has been generated elsewhere on a faster parallel machine, and then stored and analyzed at NERSC.

To continue this efficient use of resources, it is highly desirable that the size of the mass storage at NERSC scale up with the computing power. A rough scaling law for lattice QCD would be that on a dedicated machine one wants to save the entire contents of memory every two days. In the extreme case of a dedicated TeraFlop machine with a 300 GigaBytes memory, this means generating 5 TeraBytes per month. It should be noted that this is a very extreme example, not the least because it assumes that one problem area would be running in dedicated mode. However, it may not be unrealistic if the QCD TeraFlop project moves ahead, and if it is deemed desirable to store lattices and propagators at NERSC for reuse by other collaborations.

Figure B1

Appendix C—Electronic Structure, Quantum Chemistry and Chemical Reactions

I. Electronic Structure and Molecular Dynamics

At the heart of theoretical chemistry, which provides the fundamental tools to address the practical problems of the properties of biological molecules, materials, atmospheric reaction rates, solutions of environmental pollutants, and other chemical systems, is the calculation of the electronic structure of molecules. The forces between two reacting molecules, or between a molecule and a solid surface are determined by the quantum mechanical behavior of the electrons which form the bonds of molecules and determine their chemical and physical properties.

Aside from relativistic effects, which are occasionally important for heavier atoms, the physics of molecular electronic structure is determined exactly by the Schroedinger equation. More than four decades of research in quantum chemistry has resulted in the development of a mature set of algorithms and codes which implement them to solve that equation at various levels of approximation. Moreover, the experience has been accumulated which allows researchers to know what level of approximation is necessary to answer various kinds of questions about molecular interactions. The difficulty of electronic structure calculations scales with between the second and sixth power of the number of atoms depending on the level of approximation being used. New algorithms are constantly under development, including those making use of Monte Carlo methods, which scale more favorably.

We can now perform the simplest such “*ab initio*” (no adjustable or experimentally adjustable parameters) calculations, Hartree-Fock calculations, on molecules having on the order of 100 atoms. More accurate calculations, which are often needed to compute the forces between molecules, called “configuration interaction” calculations, can be performed on systems with on the order of 10 atoms.

These calculations produce the “energy surfaces” which give the forces between reacting species, between molecules and solids or between molecules and solvent molecules in a liquid solution. They provide the essential physical input to calculations of reactive scattering, including quantum effects, which yield the rates of, for example, important combustion reactions.

Current *ab initio* codes are quite large (up to hundreds of thousands of lines of code) and, over the next few years, the transition of *ab initio* calculations to massively parallel computers will require strategies that allow existing code to be implemented without the need to rewrite large fractions of the code. Because there are serial code sections in all applications, small numbers of more powerful processors will be more efficient than large number of less powerful ones. The easiest computing model for implementing existing code is the replicated-data algorithm. With this model, the major constraint on the size of problems that can be addressed is the size of the local memory. *Ab initio* calculations produce extremely large data structures (the integrals) that cannot be held in memory, and conventional approaches have stored these integrals on disk. Currently, the I/O capabilities of MPP machines is inadequate for this task and direct methods are being developed in

which the data is recomputed as it is needed. This puts additional stress on the need for fast processors to offset the inadequacy of the I/O. Finally, interprocessor communications play an important role in the efficiency of *ab initio* programs on MPP machines. All parallel applications require interprocessor communications which are affected by the bandwidth for data transfer and communication between thousands of processors. There is a need for larger bandwidths and lower latency times than are currently available.

Recent exploratory research in developing computational chemistry applications for massively parallel computers includes the implementation of the DISCO code on the Intel Touchstone Delta computer. DISCO, which was originally developed by a team headed by J. Almlvf at the University of Minnesota, implements the direct Hartree-Fock method for approximating the electronic wavefunction of molecules. At Pacific Northwest Laboratory (PNL), key sections of DISCO were rewritten by M. W. Feyereisen and R. A. Kendall to better distribute the work on one of today's massively parallel computers, the Intel Touchstone Delta supercomputer. The most computationally intensive part of the application is the calculation of the two-electron repulsion integrals; this step takes about 99% of the computational time on a serial computer. This step was parallelized by replicating the key matrix being assembled by the program (the Fock matrix) and using a master/slave procedure to perform dynamic load balancing for the integral and matrix element construction. Performing this step in parallel on 16 Delta nodes produced performance roughly equal to the code's performance on a single processor Cray X-MP. Running on all 512 processors of the Delta yielded a throughput roughly 250% higher than an eight-processor Cray Y-MP. PNL scientists are now using the massively parallel DISCO program for production work modeling solvent-solute interactions.

The standard approach to calculating accurate quantum mechanical rate constants for gas-phase reactions employs expansions in basis sets for the internal degrees of freedom in all accessible asymptotic arrangement channels. For a bimolecular reaction there will be $3N-7$ internal degrees of freedom for N atoms. For systems with more than just a few atoms, the limiting factor in this approach is evaluation of the $(3N-7)$ -dimensional integrals of the potential energy function, which scales like $M(3N-7)$, where M is the number of quadrature points needed in each coordinate. This approach is typically limited to 3 atoms. Newer methods which use a discrete variable representation obviate the need to evaluate integrals over the potential energy function, and these methods typically scale like the effort needed to diagonalize or invert a matrix of dimension $L \times L$, where L is the total number of grid points. These methods will scale like N^2 . With these new methods, calculations on 3-atom systems are routine and calculations on reactions involving 4 atoms are possible.

The simulation of the dynamics of larger systems such as biomolecules or smaller molecules in condensed phases relies on solving classical equations of motion—the Molecular Dynamics method. The computational effort in these calculations is determined largely by the complexity of the analytic functions used to represent the potential energy surface. MD simulations of biomolecules use potential functions that represent bonded interactions involving pairs, triples, and quadruples of adjacent atoms in the molecule and sums over interactions between pairs of non-bonded atoms. For large enough systems the non-bonded interactions are the most costly and scale like N^2 , where N is the number of atoms in the simulation. Most MD simulations of liquids are limited to pair-wise interactions. Important electrostatic interactions are typically incorporated

by including coulombic interactions between pairs of fixed charges at sites on the ion and in the water molecules.

For ionic solutions, simple pairwise additive potentials neglect the pole effect that ions have on the electron distribution in the solvent molecules and vice versa. In one approach, this effect is included by placing induced point dipoles at each charge site. Each dipole is equilibrated with the fixed charges and other dipoles in the system. This polarization potential is a many-body interaction (i.e., it incorporates the effects of 3-body, 4-body, and higher interactions). It is purely attractive and, like the other electrostatic interactions, is long range. To account for the repulsive many-body interactions that are more important at short range, three-body exponential repulsions are included between the ion and pairs of water molecules. The inclusion of these interaction terms provides a far more reliable model of ionic solutions than previously available, at the cost of increasing the computational requirements. However, these terms do not exhibit prohibitive scaling relations, but just increase the computational expense by a multiplicative factor (roughly an order of magnitude).

Interprocessor communications is the major limiting factor in implementing MD algorithms on MPP machines. If the problem can be broken up into running many independent trajectories, then the communications problems are minimized and the application becomes embarrassingly parallel. However, for many problems such as simulations of biomolecular dynamics, it is necessary to run one very long time trajectory. For these problems, four basic algorithmic strategies have been developed. The strategy of a master processor apportioning work to subordinate processors is the simplest to implement but puts the most demands on the communications. A replicated-data algorithm exchanges data among the processors and also puts heavy demands on the communications. The systolic-loop algorithm has the advantage of greatly reducing communications during the calculation. Another effective method is domain decomposition which apportions data to processors based upon spatial partitioning of atoms in the simulation. Although the last two methods greatly reduce communications needs, they are also the methods which require the most reprogramming of serial codes to implement.

A new MD program was written by M. A. Thompson for massively parallel computers. Performance tests were carried out on a 9,740 atom protein, giving efficiencies of 83%, 66%, and 44% respectively on 32, 64, and 128 nodes of the Intel Delta. The performance of PARADYN on 16 nodes of the Delta is roughly comparable to a single Cray Y-MP processor, and on 128 nodes, performance is 4 to 5 times that of the Cray. However, replicated-data MD approaches of this type cannot efficiently employ larger numbers of processors on today's generation of massively parallel computers. The bottleneck is the amount of time spent communicating the computed forces among the processors at the end of each time step. It is anticipated that the next generation of parallel computers will have improved interprocessor communications, permitting efficient operation of PARADYN with 3 to 4 times as many processors and correspondingly larger computational studies.

Increases in the numbers and speeds of processors will open a new domain of molecular systems and problems that can be addressed. This includes calculations of greater accuracy on currently addressed problems, calculations on systems much larger than those addressed to date, and much longer time scales for MD simulations. The theoretical maximum size of system that can be

addressed by the different types of calculations will be determined by how the problem scales. The calculations of pairwise interactions nominally scales like $N(N-1)/2$, but for sufficiently large systems (10,000-100,000 atoms depending on the form of the force field) the scaling decreases from N^2 to N . For Hartree-Fock calculations the scaling is determined by the integral calculations and assembly of the Fock matrix which nominally scale like N^4 . Once the molecule becomes large enough (approximately 100 atoms) many of the atoms in the molecule are sufficiently distant from one another that the integrals can either be neglected (set to zero) or approximated by multipole expansions. In this case the scaling becomes less than N^3 (e.g., $N^{2.7}$ in the multipole expansions). The accurate calculations of reaction energetics requires including electron correlation by methods such as configuration interaction or multiconfiguration SCF. These methods scale like N^6 , and since they are limited to relatively small molecules (10-20 atoms) no lower scaling is predicted over this range. Davidson and Feller, [Chem. Rev. **86**, 681 (1986)] give more details about scaling.

For configuration interaction (CI) or multiconfiguration self-consistent field (MCSCF) calculations of accurate energies, increases of computing power to the TeraFlop range will allow us to address systems with 2 to 3 times more atoms and to provide more accurate energetics for smaller systems. This will increase the domain of accurate calculations of rates for key chemical reactions important in areas such as combustion. Also, the study of solvent effects (cluster and bulk) on reaction energetics will require large increases in the number of atoms that can be studied. In the area of HF calculations, increases of computing power to the TeraFlop range will allow calculations on systems that are an order of magnitude larger in size than those addressed today. These problems will include biotechnology, materials, and more realistic cluster models of condensed phases. Over the next several years we should expect to see more direct combinations of *ab initio* calculations of potential energies and gradients with dynamical calculations. This combination will be a major breakthrough as it will obviate the need to parameterize force fields and it will include important effects of the reorganization of the electron distribution in strong interactions that are currently lacking in most empirical force fields.

Calculations on molecular systems important in environmental chemistry will also be impacted by increases in computing power. As an example, consider periodic Hartree-Fock calculations of total energy for a single geometry on mineral surfaces which currently require over 15 hours of CPU time, 128 Megawords of memory and over 5 GigaBytes of disk storage on the Cray Y-MP at NERSC. The desire is to study the dynamics of adsorbates and ultimately aqueous solutions interacting with surfaces of the mineral. Clearly, huge increases in computing power over the current supercomputer capabilities are needed to allow MD simulations to be driven off force fields from periodic HF calculations such as these. The electrostatic potential map spanning two layers of the mineral kaolinite is shown in Figure C1.

For MD simulations, increases to TeraFlop performance would allow increases of up to two orders of magnitude in the number of atoms that can be treated or increases of 3-4 orders of magnitude in the time scales of simulations. Increases in the size of simulations will have an impact on a number of important fields. For example, simulations of complex heterogeneous systems are important for understanding micelles, Langmuir-Blodgett films, contaminant transport in aqueous environments and at aqueous/mineral interfaces. Studies of phenomena that are determined by ionic interactions and collective effects such as phase transitions will be enabled. More realistic

Figure C1. Kaolinite electrostatic potential map

treatment of dilute solutions requires much larger simulation sizes than are currently accessible. Increases in the length of time of simulations will also help open up simulations of long time phenomena, such as protein folding, that have characteristic time scales that are orders of magnitude longer than the current picosecond to nanosecond time scales that are currently achievable.

II. New Directions for Quantum Monte Carlo Applications

Molecular modeling using massively parallel supercomputers promises to revolutionize research in chemistry and materials science over the next decade. The design and subsequent production of new chemicals tailored to replace and out perform traditional or scarce materials is a cornerstone of technological advances. Leadership in the discovery and production of new chemicals and materials can help keep the U. S. workforce competitive in the world economy.

Research in chemistry and new materials is largely based on laboratory techniques. Since the invention of electronic computers, however, it has been recognized that answers to many chemical questions could be found by solving the electronic non-relativistic Schroedinger equation. Until recently, however, the limits of computing speed, memory size, and disk space have made the approach practical only for simple molecules. Planned introduction of massively parallel supercomputers promises a major breakthrough in both the complexity and precision of problems amenable to solution by computational methods.

Quantum Monte Carlo (QMC) is one of several techniques for solving the non-relativistic electronic Schroedinger equation. It has demonstrated accuracies for the computation of molecular energies rarely achieved by other approaches, and has distinct advantages in massively parallel computing environments. For all but the smallest systems, solving the Schroedinger equation inevitably requires the introduction of approximations. The essence of the QMC method is to solve the Schroedinger equation stochastically by sampling from a distribution that can always be chosen to be an improvement over the sampling equivalent to any variational calculation. The precision of the method is then limited only by the size of the computed sample.

The power of the QMC method is that it is inherently an N-body method that can capture all of the instantaneous correlations of electrons, including dynamical correlations. These require extensive, slowly convergent, configuration interaction expansions that are rarely possible to carry out using other methods because of computer limitations. The QMC approach yields accurate energies, including binding energies, barriers to chemical reaction, and electron affinities, from the current generation of supercomputers, and has been used to determine other properties, including dipole and quadrupole moments, oscillator strengths, and molecular geometry energy gradients. QMC methods have been used to calculate atomic and molecular properties of, for example, Cu and Fe (pseudopotential methods), LiH , H_2 , N_2 , and CH_4 (all-electron methods). Results are in excellent agreement with experiment.

QMC computations are ideally suited to highly parallel computer architectures. They require very little in terms of memory or disk I/O. Execution time scales as the computed sample size and the complexity of the model. Precision increases as the square root of the sample size. Increases in computational speed may thus be used to increase precision, the size of system, and the complexity of the model.

QMC computations have already been ported successfully to both vector and massively parallel supercomputers. Thus QMC can use a massively parallel machine extremely effectively from day one of operation.

The following table summarizes typical increases that might be achieved over the next three years using massively parallel supercomputers with the indicated sizes and speeds.

<i>Year</i>	<i>1993</i>	<i>1994</i>	<i>1995</i>
Number of processors	100	1000	10000
CPU speed (MF per processor)	100-150	300-600	600-1000
Model complexity increase	2	60-80	1200-2000
Precision increase	1.4	10	40

If systems with the indicated computing power are delivered, 1995 will see computational estimates of the binding energy and other properties of molecular systems (molecules, clusters, and larger aggregates) containing greater than 100 active electrons, i.e., those outside of closed shells. This capability will make possible the computation of a wide range of organic, organometallic, and inorganic systems yielding results rivaling the accuracies that have been achieved for the modest-size systems treated to date. Such a development would represent a major breakthrough and would inevitably foster new insights, applications, and products.

Appendix D—Molecular Biology

The outstanding problem in the arena of nucleic acid and protein structure determination is the reliable computational location of the static minimum free energy structure in the natural environment of the cell, i.e. in aqueous milieu in the presence of appropriate concentrations of salts, and the calculation of the molecule's dynamic trajectory in space and time at room temperature. The computational approach is important because experimental results are still so difficult to achieve. X-ray crystallography, which yields atomic resolution views, relies on the unquantified art of growing crystals, followed by the laborious process of translating the diffraction pattern to an electron density map. High resolution nuclear magnetic resonance, which probes the dynamic structure in solution, does not itself produce atomic resolution views. The scanning tunneling microscope, which does, in principle, produce atomic resolution views, does not employ a natural environment.

Two computational approaches to structure determination currently employed are energy minimization and molecular dynamics simulations. Energy minimizations yield static structures, and they typically treat the environment as well as the entropic contribution to the free energy only crudely. Molecular dynamics simulations, with explicit solvent and salt, produce time and space dependent trajectories from a chosen stationary starting state (usually a crystal structure, at present). Both of these methods are currently severely hampered by lack of adequate computing power, since they are both highly computation intensive.

Energy minimization suffers from the well-known multiple minimum problem, and strategies to cope with it have been devised. One such method, known as the build-up technique, has been successfully employed in the prediction of polypeptide structures. The technique requires global scale searches of the conformation space of the amino acid building blocks, followed by combinatorial build-up of the building block energy minima to generate larger structures a residue at a time. A recent pilot project has revealed that the method can also be successfully applied to polynucleotides. However, the combinatorial build-up to larger structures is much more computation intensive for polynucleotides than for polypeptides because many more conformational degrees of freedom must be simultaneously optimized. Indeed, current constraints limit the number of duplex residues that can be built combinatorially to three, since the CPU time required per trial at present increases as n^2 with the number of residues, and the technique, being combinatorial, requiring thousands of trials to add each residue. Massively parallel computers, capable of carrying out thousands of calculations at the same time, are needed for effective employment of this technique. Structures thus obtained, without any experimental input other than the force field parameters, can then serve as starting conformations for molecular dynamics simulations with explicit solvent and salt.

Unfortunately, however, computing constraints also strongly limit the utility of molecular dynamics simulations at the present time. The simulations now are typically confined to the 100 picosecond range, while biologically and chemically interesting structural movements may take place in time frames that are 10^4 times longer. In addition, one would wish to study much longer pieces of DNA than the duplex 12-mers that are usually investigated at the moment, and to study the DNA

in association with the natural proteins that govern the DNA function. A current dream is to accurately model by molecular dynamics simulations such natural processes as DNA replication, a highly dynamic activity involving the DNA and more than a dozen proteins. Such a simulation could show how a mutation results that initiates cancer when DNA modified by carcinogenic chemicals replicates.

Finally, one would ideally wish to observe graphically, in real time, and in three dimensions (via a holographic movie on one's local workstation) the dynamics simulation as it is being computed. This would permit instant visual comprehension of the results.

The above problems associated with structure prediction are of particular importance in the Human Genome Initiative. The Human Genome program will surely be expanding into the realm of molecular structure, since a detailed understanding of the functions of unique sequences cannot be achieved without an understanding of their underlying structures. For example, certain DNA sequences which possess features of shape that are hallmarks of their function include origins of replication, which are bent, and telomeres—the ends of chromosomes—which form multiple strands. Accurate computational prediction of such structures from sequence would permit screening the mass of incoming sequence data that is non-protein coding for function, occurrence and distribution. In addition, computational prediction of protein structures will also be needed, particularly to elucidate how the structures of mutated proteins that cause human disease differ from their normal counterparts. Furthermore, the Human Genome Initiative will pose additional computational challenges due to the large amounts of DNA sequence data that will be generated. This data will require large-scale computing both to store the sequences, and to locate specific genes, control regions, repetitive or other DNA sequences of interest, from the enormous mass of bases delineated. Many stepwise sequence comparisons over millions of nucleotides will be required. Clearly, advanced parallel processors will be essential for the task, or perhaps hard-wired, dedicated machines.

Appendix E—Plasma Physics and the Numerical Tokamak Project

I. Numerical Tokamak Project (NTP)

Over the past two decades, the fusion power produced by experiments has increased by over seven orders of magnitude; yet a fundamental problem remains unresolved, namely how to design and operate fusion reactors that minimize losses of particles and energy. Since 1988, the DOE has focused on this issue, and the Numerical Tokamak Project (NTP) represents an aggressive program to combine the recent advances in theoretical understanding of plasma physics with the powerful supercomputing platforms (in particular massively parallel computers) just emerging to prominence.

The computational program focuses on the plasma turbulence now currently believed to be responsible for the transport loss effects. The difficulty is to be found in the attempt to gain a computational grip on the very broad range of space and time scales exhibited by a plasma evolving in a tokamak. The issues to be addressed if meaningful results are to be extracted include true toroidal geometry, nearly collisionless motion of particles as they move along their field lines, and the disparate time scales of particles as they move parallel and perpendicular to the magnetic field. Further, electron effects cannot be neglected. Electrons, due to their relatively light mass, tend to move at a much higher velocity than the ions and therefore behave differently in the device. The computational models must respect the smallest space scales (which might be related to the gyro-radius and be no more than a few millimeters) and the largest (which are on the order of the major radius).

To reliably address such a mix of issues, two principal approaches are being followed and results are being compared. In the most fundamental approach, a collection of particles is being computationally advanced to sample the motion of actual plasma particles. The resulting particle data is used to calculate plasma currents and charge densities by interpolating to a spatial grid and finally solve Maxwell's equations for the self-consistent electromagnetic fields. This particle approach is faithful to the physics, but its numerical resolution is limited. In an alternative approach, a set of fluid equations is used which deals with parallel and drift motions by advancing a few velocity moments of the particle velocity distribution with approximate models for the higher moments. Maxwell's equations are again solved to complete the self-consistent description. Currently, solutions from these two methods are being compared with one another, with the predictions of analytic theory, and with experimental observations.

The NTP is adapting both approaches to provide two independent numerical treatments in a region where the times scales overlap. If validated in the overlap region, fluid models, which require many fewer degrees of freedom than particle models, may provide a much more efficient means of studying transport. Comparison of the fluid and Particle-in-Cell (PIC) approaches will also quantify several issues related to the numerical precision of the latter, including how the shot noise of individual particles affects the numerical solutions.

(a) Gyrokinetic Particle-in-Cell Methods

The outstanding development in advanced kinetic algorithms is the gyrokinetic particle simulation model. This model formally averages time over the cyclotron time scale while retaining spatial variations on the length scale of the Larmor radius. The validity of the gyrokinetics formalism depends on an ordering scheme in which temporal variations are slower than the cyclotron frequency, spatial variations in the equilibrium quantities are weaker than those in the perturbing electromagnetic fields, and velocity perturbations in the fields are smaller than the thermal speed. This set of assumptions is well satisfied by a variety of instabilities that occur in the core of tokamaks and are believed to be the principal agents of anomalous transport. In not being constrained to the cyclotron time scale, the simulation can employ a larger timestep.

Nonetheless, current gyrokinetics codes, which are running with limited physics models and relaxed parameters, still overpower current Cray supercomputers when the attempt is made to calculate the effects of turbulence in even a small subsection of a tokamak. For example, it is reported in "Comprehensive Numerical Modelling of Tokamaks," (UCRL-ID-105650) by R. Cohen et al. that simulations of ion temperature gradient modes and trapped electron modes done by Sydora require about 100 hours on a Cray-2. They also report that partially linearized simulations utilizing adiabatic electrons have allowed some 3D simulations to complete in less than an hour, although these assumptions limit the physics. Some of these codes have been ported to the CM-2 (by Princeton, a 2 1/2D electromagnetic code with partially adiabatic electrons, and by LLNL in collaboration with Sydora, a 3D electrostatic code). Another LLNL effort is that of currently designing a gyrokinetics code on the BBN TC2000, and this code will eventually be ported to a more powerful MPP supporting shared memory and data parallel programming models.

It is reported in Cohen et. al. that these Cray simulations have made gross simplifications, from utilizing artificially large gradients to extremely crude meshes, to obtain numbers with the computational resources available. Realistic parameters would require at least a thousand fold increase in throughput. An estimate is presented which argues that a TeraFlop machine might require 300,000,000 hours to simulate an energy confinement time for large machine. However, it is also reported that a number of reasonable simplifications can be made which reduce the computer time requirements, while continuing to extract useful physics. First, implicit techniques might allow the program to run on the ion time scale, which might reduce the compute time to a mere 20,000 hours. However, the authors make the point that if the simulation seeks only to resolve the turbulence times scale for the purpose of determining transport coefficients or fluxes, then an implicit code might succeed in about 20 hours. Finally, utilizing linearized simulations could reduce the compute time by another two orders of magnitude, to approximately 0.5 hours on a 300 GigaFlop machine.

The authors argue that such a calculation would be most useful in a hybrid simulation, in which the gyrokinetics module and both 1- and 2-D transport codes are run in tandem, the PIC code computing the fluxes when necessary and the transport codes advancing the relevant macroscopic variables in the physical regions where each is pertinent. The PIC code would be called only when the profiles produced by the transport codes have changed significantly. This kind of simulation would model the entire tokamak for transport time scales, and is a possible candidate problem for a 300 GigaFlop MPP, but not for a 16 GigaFlop C90. Note that in an intermediate stage, the

transport codes could utilize a C90 while the PIC code would use the MPP. The kind of problem envisaged by the authors would add numerous modules, including neutrals, RF etc., so that an accurate representation of tokamak operation could be realized. Finally, it is observed that the memory requirements of such a run, while great, are not overwhelming. The memory required to model a moderately large tokamak was put at about 100 GW, which is well within the range of capability of 1994 offerings.

The above represents the kind of ambitious problem which is at least explorable on a 1993-1995 MP platform; at a minimum with such a device, one could continue the development of such a modularized calculation, and each piece would fit on the computer, could be debugged there and could be run at reasonable resolution for periods of time on the order of an hour, allowing the researcher to begin to test and evaluate his model.

(b) Fluid Approach to the Numerical Tokamak

The second modeling area involves the use of gyrofluid equations. Traditional fluid equations, such as those of magnetohydrodynamics (MHD) (discussed below), have had a long and successful history of explaining large-scale equilibrium and stability properties of plasmas. However, they lack many kinetic effects (such as Landau damping) that are important for dealing with instabilities of the lower-frequency, shorter-wavelength, drift type, believed to be responsible for plasma turbulence. The particle methods described above treat such instabilities automatically; only recently has a method been found to approximate them in fluid equations. This is an important advance over traditional theory. It helps establish consistently the spectral energy density of the saturated turbulent state, from which the transport properties can be determined.

The collisionless mixing of particles is related to the Landau damping phenomenon often seen in plasmas. To model this process in fluid equations, we typically add a parallel heat conductivity to the temperature evolution equation to provide a damping rate of the order of the collisionless thermal mixing rate (the magnitude of the parallel wave number times the thermal speed). This can easily be evaluated with Fourier methods and turns out to be a fairly good Pade, or multipole, approximation to the linear kinetic plasma response. This procedure converges fairly rapidly if higher fluid moments are kept before introducing this dissipative closure approximation. When self-consistent electric fields are added to this model, it describes not only Landau damping (which provides an important energy sink for nonlinear turbulence simulations) but also inverse-Landau damping, which drives some types of plasma instabilities.

Another important kinetic effect is that an ion responds not to the local electromagnetic field but to the field averaged over the ion's gyro-orbit. We have recently developed Pade-like approximations that model the linear and nonlinear consequences of this effect. Such models enable the formulation of parabolic problems that can be solved by using fast Fourier transforms either alone or in combination with tridiagonal solvers.

While current computer technology and numerical algorithms are deemed adequate to perform turbulence calculations using fluid models for edge plasma parameters, the next generation of massively parallel mainframes with near-TeraFlop performance, combined with new algorithms and software development, are needed to tackle the fluctuations at the core of present-day

magnetic fusion devices such as the tokamak. Formidable analytical and computational difficulties are presented by the toroidal nature of the geometry and the ability to model hotter plasmas with increased resolution in space and time scales, either by adopting a fully kinetic approach, by extending the fluid-like models to include relevant kinetic effects or a combination of both.

The possibility of deriving fluid equations with closure relations compatible with Landau damping, which also include ion finite Larmor radius (FLR) effects, has broadened the regime of applicability of the fluid models to the plasma core. These so-called gyrofluid models have gone a long way toward alleviating some of the analytical difficulties. In order to overcome the computational difficulties, a major effort at ORNL is being undertaken to implement fluid turbulence models on existing massively parallel computers. These models use a spectral representation in poloidal and toroidal mode numbers and finite differences in the radial coordinate. The nonlinear terms lead to mode convolutions that are performed analytically, rather than using Fourier transforms.

Preliminary tests with a reduced resolution fluid model were indicating that performance comparable to a single processor of the Cray-2 could be achieved with 32 processors of the ORNL Intel iPSC/860. In a collaborative effort with NERSC staff, similar performance was achieved on LLNL's BBN TC2000. Recently, optimization with the new Portland Group compiler PG-03 and full use of the 128 processors on the iPSC/860 by-passing the host, have yielded performance 5 times that of a single processor of the Cray-2. This is slightly better than the factor of 4 achieved to date on the C90 as compared to the Cray-2. More importantly, our recent performance studies on the iPSC/860 show that varying the number of modes retained in the fluid calculations changes the optimal number of processors for comparable timing per step. Not surprisingly, more modes entail a greater optimal number of processors. This scalability has been confirmed on the Caltech Intel Delta where use of all 512 processors has resulted in an order of magnitude improvement over the Cray-2.

While these measures of performance are encouraging, the largest calculations performed to date have involved 1000 radial grid points and 512 modes, mainly because of the limitation to 16 MegaBytes of memory per processor. These calculations are of factor of 4 shy of the minimum resolution required, with 1000 grid points and 2000 modes, to accurately describe the fluctuations at the core of the TEXT and TFTR tokamaks. We expect to be able to routinely carry out such minimum resolution calculations on a 200-400 GigaFlop MPP, not only because of 100 plus MegaBytes of memory per processor on the MPP, but also because a conservative extrapolation of our iPSC/860 and Delta scalings to the MPP indicates that a two order of magnitude improvement over a single processor of the Cray-2 is to be expected for the ORNL models.

As the computational mesh refines, however, the numerical error will drop below the error associated with assumptions made in deriving the physical model itself. For example, most current gyrofluid computations make a number of simplifications out of convenience: the calculations are generally electrostatic and generally ignore electron dynamics. Electrons are treated adiabatically, which is to say that the ions determine the electrostatic potential and the perturbed electron density arranges itself quasistatically accordingly. A gyrofluid simulation of ion temperature gradient-driven turbulence in a tokamak is shown in Figure E1.

Figure E1. Gyrofluid simulation of ion temperature gradient-driven turbulence

Finite electron inertia and parallel current may well be quite important in determining the turbulent state, but the added complexity of the model including these effects and the added computational difficulty has thus far discouraged development in these areas. In particular, the electron modes have much longer wavelengths (at similar frequencies) owing to their higher thermal speed, and this presents a difficult problem for spectral algorithms. To accurately represent the small wavenumber electron modes and the high wavenumber ion modes using current algorithms requires much higher resolution meshes. Simply using a faster computer may not solve this problem—new algorithms need to be developed to bridge the gap. A collaborative effort between LLNL and GA is exploring a number of possibilities to tackle this problem, including multi-scale spectra. With the availability of a 100-300 GigaFlop parallel computer, it becomes possible to contemplate the addition of some of these additional physical effects. Thus, it is hoped that some combination of finer resolution and enhanced physics will allow a much more detailed computational picture to evolve on MP platforms in the 1993-1994 time range.

The other side of the coin, trying to bring the analysis, model development and assessment, execution and improvement cycles into a realistic daily working schedule, involves using more powerful computer-supported collaborative work communications and analysis tools. A computer environment which supports generating and storing enormous quantities of data, but lacks the powerful data navigation and visualization tools necessary to determine quickly and effectively whether the data should be archived or destroyed, soon becomes dysfunctional. Interactive access to data with such tools should be supported at the highest level.

II. Particle-in-Cell Calculations

The Particle-in-Cell (PIC) method is a widely used technique for modeling ionized plasma behavior. The beauty of this method, in which a relatively small number of macroparticles are used to represent the kinetic behavior of a much larger set of real particles, is that it provides not only a useful, correct description of the physics, but also some hint of the underlying stochastic nature of the plasma. The intuition that follows from sorting the signal from the PIC noise coming from a relatively small number of simulation particles is in many ways very similar to the associated experimental situation that the simulation models.

The problem is that the most accurate simulations are the ones that make the fewest assumptions. In the PIC simulation world, this translates into explicit integration of the particle orbits and Maxwell's equations. This model, useful for all sorts of plasma simulations from microwave cavities to beam propagation, exhibits not only the ability but, unfortunately, the requirement to follow detailed electron motion on the fastest time scale. Often the details of electron motion are not required and many models have been formulated to average over these high frequencies, but the fact remains that these codes are at best approximations of (and frequently only a small factor cheaper than) the better physics that is available from full explicit simulation.

State-of-the-art PIC computing in 1992 has progressed to the point that very realistic 2.5-D (5-D phase space) fully relativistic, fully electromagnetic simulations can be routinely accomplished with on the order of 10 CPU hours on the large mainframes. These codes are now being ported to

some of the more capable workstations. The state-of-the-art for fully 3-D calculations is not as well developed. Even on the largest and fastest present-day machines, 3-D (6-D phase-space) calculations require more thought about data structure manipulation than about the basic physics models that can be comfortably run. For example, explicit 3-D relativistic, electromagnetic simulations of microwave generators typically take 100 CPU hours to simulate a few microseconds of real time. Typically such simulations suffer limitations on the computational grid resolution; lower limits are set on resolution by the requirements that a light wave not travel more than a cell in a time step. The finer the resolution the smaller the time step.

Even very optimized, special purpose codes such as the heavy-ion-fusion ion transport code WARP, that is nonrelativistic and electrostatic, is still hampered by constraints that the fastest particles not move more than a cell in a time step—leading again to runs of over 30 hours for typical studies of ion transport in the middle section of the device. Simulations encompassing the detailed resolution of the source and the beam gymnastics in the final focus are still beyond the scope of anything less than heroic runs. Studies of sensitivities to geometric configuration or other parameters, the real cost-saving utility of simulations over experimental programs, are simply not feasible today.

In the 1993-1995 time frame, we may well expect massively parallel computational facilities with the capability to expand our computational capabilities by perhaps a factor of 100. In addition to making the two previously mentioned examples more tractable, we are eagerly awaiting machines that allow not only the needed special resolution (in spite of the more restrictive time step), but also the ability to include much more physics and chemistry. In the ion source business, tracking not only the geometric complexities but also the various concentrations of the multitude of chemical constituents as a function of time are essential to understanding and predictive capabilities. Such issues also must be addressed in the realm of plasma processing, a field that is poised for rapid growth in the next several years and one that is especially well suited to optimization via numerical simulation.

III. Magnetohydrodynamics (MHD)

MHD has proved itself a useful tool for analyzing the global equilibrium and stability characteristics for essentially all magnetic confinement fusion devices. It is used extensively by design teams to evaluate new devices (such as ITER) as well as by plasma physicists to aid in the understanding of plasma stability. There are three different types of calculations that are generally done. Equilibrium (or quasistatic) calculations are generally done in two dimensions and require elliptic PDE solvers. Iterative matrix solvers (such as SOR, line SOR, multigrid and conjugate gradient schemes) are required. Some of these calculations, particularly those which include the coupling of the plasma to external coils, can get expensive (tens of hours of Cray time). Linear stability analyses are usually done with eigenvalue solvers. These calculations are not as expensive and may not require supercomputers, unless one intends to cover a lot of parameter space. Three-dimensional nonlinear initial value codes are the most computationally expensive. Many of these codes are implicit and require sparse matrix solvers. The speedup that can be achieved by using large timesteps with an implicit algorithm is generally much larger than the speedup one would expect by using a massively parallel machine with an explicit algorithm. FFTs are also widely used in these simulation codes.

In the past, most nonlinear simulations have focused on long wavelength phenomena. The only reason these simulations are tractable on present generation supercomputers is that these instabilities tend to be dominantly quasi-linear or involve only a small part of the spectrum (i.e. the physics is not fully developed turbulence). However, understanding of transport processes is emerging as the single most important theoretical issue in the Magnetic Fusion program. From an MHD context this generally means shorter wavelength phenomena are more important and must be properly simulated. These instabilities often saturate by cascading the energy to the dissipation scales, much like standard fluid turbulence. Unfortunately, plasmas have very little dissipation and the spatial scales for the drivers and dissipators are vastly different. We believe that massively parallel machines can make a significant difference for these types of problems. Physics beyond standard MHD also significantly affects these types of instabilities which further complicates the issue by adding disparate time scales to the already disparate spatial scales.

It is very important that the massively parallel machine chosen be able to handle implicit algorithms; the basic requirement then is that the vendor offer mathematical libraries which include the necessary linear system and eigenvalue problem solvers which are typically used by these MHD codes. If these routines are to be efficient, fast global communications is a must.

Appendix F—Climate Modeling

I. Earth Systems Modeling

Modeling plays a vital role in Earth Systems science, in part because the scope and complexity of the earth system renders it inaccessible to controlled experimental manipulation. While laboratory and field study can shed light on individual processes of interest, the complex interaction of multiple processes that determine the behavior of the earth system can only be addressed through simulation. A better understanding of earth system behavior is needed as one of the important inputs to comprehensive energy, environmental, and economic policy formulation

At present, there are models of some of the major components of the earth system, such as atmospheric and oceanic dynamics and thermodynamics. Other components are under development, such as global-scale models of terrestrial ecosystem dynamics and marine biogeochemistry processes. For the most part, the existing component models have been developed independently and in relative isolation. What is required for comprehensive earth system modeling capability is the full coupling of all major component subsystems to properly reflect the realistic interaction among them. This is a daunting prospect, because each of the components in its present form constitute a major computational challenge for vector supercomputers. Furthermore, each component requires enhanced physics process representations and refined spatial and temporal resolution, requiring in turn more computational throughput.

As an example, current-generation atmospheric general circulation models (AGCMs) typically have a horizontal resolution of about 4 deg x 5 deg, and are integrated for a few decades of simulated time with a time step size on the order of a few minutes. This typically requires on the order of 200 CPU hours on vector supercomputers. The desired resolution for this system is actually on the order of 1 deg x 1 deg. What is more, when the atmosphere is coupled to other earth system components with longer time scales (e.g., ocean and terrestrial ecosystem modules), the desired simulation time scale becomes on the order of centuries. To enhance resolution, increase simulation duration, incorporate refined process representations (e.g., improved cloud-radiation interaction), and fully couple all major earth system components into the next-generation model clearly will require TeraFlop performance levels. Figure F1 shows a 2-D domain decomposition of the atmosphere into regions to be handled by each processor of a parallel computer.

However, before such a computation can even be designed, much can and must be learned about the behavior of coupled versions of current-generation models. The basic components of a climate systems model are illustrated in Figure F2. It is in this context that the utility of systems with on the order of 350 GigaFlop comes into play. With such a system, it would be possible to undertake pioneering experiments involving the fully-coupled earth system model, constructed by linking present-generation subsystems (atmospheric and oceanic GCMs, newly-developed global ecosystem dynamics models, and biogeochemistry models). This is an essential step on the path to the next-generation simulation capability, because the increased knowledge gained from experiments with the coupled system will provide guidance on how best to improve the fidelity of process representations and spatial resolution in each component.

Figure F1. Atmospheric domain decomposition

Figure F2. Basic components of climate system model

For example, early coupled model experiments with coarse models will shed light on the long-term natural variability of the ocean-atmosphere-land-biosphere system, and on the memory time of the system for initial conditions in the "slow" components. This information is needed before comprehensive, high-resolution experiments on global change can be devised, because the natural variability of the unperturbed state must be known before the statistical significance of any possible "greenhouse signal" can be determined. Multiple runs with such models must be executed to determine the sensitivity of the long-term behavior of the system to initial conditions in deep ocean circulation and biomass geographical distribution.

Finally, it should be noted that the data volumes associated with these types of experiments are truly enormous. We are dealing with the statistical properties of 3-D coupled chaotic dynamical systems with widely disparate space and time scales, integrated in time with essentially explicit time advance algorithms. The need for advanced data handling and scientific visualization facilities is perhaps nowhere more acute, among the computational grand challenges, than in comprehensive earth systems modeling.

II. Spectral Methods for General Circulation Models

General Circulation Models (GCM) are rarely run at resolutions finer than 400 kilometers when used for climate modeling on current vector supercomputers. As an example of the current limitations, the European Center for Medium-range Weather Forecast uses a model which, when run at a 100 km resolution, would require four years to complete a 100 year simulation on a Cray-2. When one considers that a coupled ocean-GCM calculation might require resolutions in the 40 km range to be run for a simulated 1000 years, the magnitude of the problem becomes evident and the current interest in scalable parallel algorithms that could eventually be utilized on super-Tera-Flop platforms becomes understandable.

The spectral transform method is widely used because of its computational efficiency. Instead of advancing the prognostic variables themselves in time, this approach instead expands quantities such as the vorticity and velocity field divergence in a truncated series of its spectral coefficients, a double Fourier-Legendre series. It is these spectral coefficients which are advanced in time. The calculation is characterized by numerous FFT and inverse FFT operations as well as Legendre and inverse Legendre transforms which are computed using Gaussian quadrature, which requires $O(n^2)$ operations as opposed to $O(n \log(n))$ for the FFT. Thus the computational cost of this approach is dominated asymptotically by the Legendre transform, which when integrated over the complete calculation translates asymptotically into an $O(n^3)$ computational cost.

Parallel implementation of the spectral method is accomplished by either a 1-D or 2-D decomposition. The 1-D decomposition is of limited interest because of its high communications overhead and because it can utilize no more processors than half the meshpoints in the latitudinal direction, J , (North-South). This limits application to small MP platforms. The 2-D decomposition is preferable. Not only does it require less communication, it can utilize up to $I(J/4)$ processors where I and J are the two grid resolutions. Typically we take $I = 2J = 3.2n$, where n is the truncation number in the series representation.

If applied to the shallow water equations, it can be shown that reasonable efficiencies (75%) can be realized on current computers (such as the Intel Delta) for high resolutions ($n=200$) in the 500 processor range. The Intel Paragon (a machine available in 1993) would show an efficiency on the order of 75% for 1000 processors in the $n=200$ range. However, as the resolution decreases, the efficiency drops off rapidly due to the rising communication requirements of the spectral method as the number of processors is increased. The throughput (which can be described as the simulated years per real-time hour) can also be estimated. The difference in capabilities between the Delta and the Paragon become manifest with this comparison: an $n=200$ calculation on a 512 processor Delta would be capable of simulating 4 years/hour, whereas a 2048 node Paragon would simulate over 20 years/hour.

For the more general GCM calculations, the situation is helped somewhat by three additional factors: the full primitive equations, the multilayer representation of the atmosphere, and the physics. These increase computation requirements considerably, but communication overhead is not increased commensurately. The result is that the scalability of spectral transform GCMs is limited by the transforms used to compute the dynamics. At low resolutions, the additional physics helps the situation, but at high resolutions, which are required for accurate climate calculations, the additional physics no longer contributes much to the overall cost due to the $O(n^3)$ effect.

Some estimates for GCMs can be presented. To get a sense of the kind of improvement one could expect in resolution from different categories of machine, we will compare the resolution at which one could run a problem given a desired throughput, for the Intel Delta (512 nodes), PARAGON (2048 nodes) and a follow on TeraFlop device utilizing 2048 nodes (computing at a rate several (~ 5) times faster than the PARAGON). Assuming a desired throughput of 0.2 years/hour, the Delta could accommodate $n=80$, the PARAGON $n=150$ and the TeraFlop device $n=200$. While the TeraFlop device clearly provides far superior resolution, we can get a sense of how far we have to go from the following estimate. An $n=185$ resolution will provide a 100 year simulation in a week! The Delta could achieve this simulation only with $n=47$ and the Paragon with $n=100$. This estimate is based on an optimistic analysis, neglecting a number of effects, including some communication requirements, potential computational and communication demands of grid-space coupling, I/O, coupled ocean, chemistry and biosphere models. Further, the analysis is done on the basis of a fixed-time step which could be unrealistically large.

Thus, at high resolutions, these long term calculations are unrealistic on even TeraFlop computers. Nonetheless, at lower resolutions, both dynamics costs and parallel efficiency are reasonable, and the additional resolution permitted by the 1993-1995 MP architecture permits more realistic simulations than would be possible on, for instance, a Delta class computer. For short term simulations where there is adequate computational power in the 1993-1995 MP architectures to handle the physics at fairly high resolution, the dominating factor will be the $O(n^3)$ computational cost of the spectral transform. The high computational requirements of parallel GCMs suggests that adaptive refinement methods might be a suitable subject for further research. Another issue of interest would be the effect of I/O on parallel performance.

Glossary of Acronyms and Abbreviations

ACL	Advanced Computer Laboratory (at Los Alamos National Laboratory)
AGCMs	Atmospheric General Circulation Models
ASTA	Advanced Software Technology and Algorithms
AVS	Application Visualization System (vendor-supplied computer graphics code)
Byte	8 bits
CFD	Computational Fluid Dynamics
CRUs	Computer Recharge Units
DELTA	Model of Intel Supercomputer
DISCOS	Name of a software vendor, a division of General Atomics Incorporated
ER	Energy Research
ERSUG	Energy Research Supercomputer Users' Group
EXERSUG	Executive Committee of the Energy Research Supercomputer Users' Group
FFT	Fast Fourier Transform
FLR	Larmor radius
FTE	Full-Time Equivalent
GA	General Atomics, Incorporated
GAUSSIAN	A vendor-supplied quantum chemistry code
GCMs	General Circulation Models
GigaBytes	1 billion (10^9) Bytes
GigaFlops	1 billion (10^9) floating point operations per second
HF	Hartree-Fock
HPCRC	High Performance Computing Research Center
I/O	Input/Output
IPE	Initial Production Environment
ITER	International Thermonuclear Experimental Reactor
MCSCF	Multiconfiguration Self-Consistent Field
MD	Molecular Dynamics
MegaBytes	1 million (10^6) Bytes
MegaFlops	1 million (10^6) floating point operations per second
MHD	Magnetohydrodynamics
MIMD	Multiple Instruction Multiple Data
MP	Massively Parallel
MPCI	Massively Parallel Computing Initiative (at Lawrence Livermore National Lab)
MPP	Massively Parallel Processor
NAG	Numerical Algorithms Group, suppliers of a mathematical subroutine library
NASTRAN	A vendor-supplied structural mechanics code
NERSC	National Energy Research Supercomputer Center
NQS	Network Queuing System

NTP	Numerical Tokamak Project
ORNL	Oak Ridge National Laboratory
PARADYNE	Parallel Version of DYNA 3-D, a structural mechanics code
PE	Processing Element
PIC	Particle-in-Cell
PNL	Pacific Northwest Laboratory
QCD	Quantum Chromodynamics
QMC	Quantum Monte Carlo
SCF	Self-Consistent Field
SIMD	Single Instruction Multiple Data
SLATEC	Mathematical subroutine library originated by Sandia - Los Alamos - Air Force Technical Exchange Committee
TeraBytes	1 trillion (10^{12}) Bytes
TeraFlops	1 trillion (10^{12}) floating point operations per second
UNICHEM	A vendor-supplied quantum chemistry code

END

**DATE
FILMED**

8/1/71 93