The Office of Science Data-Management Challenge

Report from the DOE Office of Science Data-Management Workshops

March–May 2004

Contents

Preface

In June 2004 the DOE Office of Advanced Scientific Computing Research held a meeting to discuss the ASCR/MICS Strategic Plan. A number of the application scientists and computer scientists at the meeting came to the vocally expressed conclusion that the plan as presented was dangerously light on attention to data management, given the increasingly data-intensive nature of research supported by the Office of Science. This constructive criticism was well received and resulted in encouragement to hold a series of workshops that would be able to document gaps between the needs of application sciences and the data-management technology and tools expected to be available.

The first workshop was held at SLAC on March 16–18, 2004, focusing on understanding application-science needs and currently available technologies. A smaller meeting of the "Extended Organizing Committee" was held at SLAC on April 20–22, 2004, discussing how to structure the workshop report and the program of the final workshop. The final workshop was held in Chicago on May 24–26, 2004, with a focus on understanding commonalities of need and on quantifying and prioritizing the costs of meeting the needs. After the final workshop, a series of phone conferences, open to all workshop participants, reconciled the many simultaneous writing and editing efforts.

The workshops were far from being "yet another workshop to document needs of which we are all already aware." The essentially unanimous opinion was that the workshops were exciting and valuable and advanced many participants' thinking on data-management issues. Of particular value was a "revolt" by some application scientists at the first workshop—a revolt provoked by being asked to consider the value to their work of apparently obscure computer science issues. For example, the word "ontology" was outstandingly successful in generating apprehensive incomprehension. Fortunately, the immediate outcome of the revolt was a successful attempt to reach a common understanding of the real issues facing scientists whose work has only recently become data intensive.

The program of the workshops and the majority of the presentations are available at http://www-conf.slac.stanford.edu/dmw2004

Richard P. Mount

November 30, 2004

Acknowledgments

The first workshop and the Extended Organizing Committee meeting were held at SLAC. Arla Lecount, Jane Rochester, and Jennifer Huang-Le assured the logistical success of these meetings.

The final workshop in Chicago relied heavily on Argonne staff, most notably Angel Murakami.

Support for the Web site was provided by David Lee, and Brooks Collins took charge of making talks at the SLAC workshop available on streaming video.

Gail Pieper edited the report, showing an uncanny understanding of the issues and ruthlessly exposing inadequate logic.

Terry Anderson designed the report cover.

Within the MICS Office, John van Rosendale ensured financial support for the workshops and gave valuable advice on the program and on participants to be invited.

Finally, thanks are due to all the participants, and especially those from other agencies and from industry, who made the workshop series a productive activity for the Office of Science—productive even in advance of any increased financial support for data management.

Richard P. Mount

November 30, 2004

Executive Summary

Science—like business, national security, and even everyday life—is becoming more and more data intensive. In some sciences the data-management challenge already exceeds the compute-power challenge in its needed resources. Leadership in applying computing to science will necessarily require both world-class computing and world-class data management.

The Office of Science program needs a leadership-class capability in scientific data management. Currently two-thirds of Office of Science research and development in data management is left to the individual scientific programs. About $18M/year is spent by the programs on data-management research and development targeted at their most urgent needs. This is to be compared with the $9M/year spent on data management by DOE computer science. This highly mission-directed approach has been effective, but only in meeting just the highest-priority needs of individual programs. A coherent, leadership-class, program of data management is clearly warranted by the scale and nature of the Office of Science programs. More directly, much of the Office of Science portfolio is in desperate need of such a program; without it, data management could easily become the primary bottleneck to scientific progress within the next five years.

When grouped into simulation-intensive science, experiment/observation-intensive science, and information-intensive science, the Office of Science programs show striking commonalities in their data-management needs. Not just research and development but also packaging and hardening as well as maintenance and support are required. Meeting these needs is a medium- to long-term effort requiring a well-planned program of evolving investment.

We propose an Office of Science Data-Management Program at an initial scale of $32M/year of new funding. The program should be managed by a Director charged with creating and maintaining a forward-looking approach to multiscience data-management challenges. The program should favor collaborative proposals involving computer science and application science or, ideally, multiple application sciences. Proposals bringing substantial application science funding should be especially favored.

The proposed program has many similarities to the DOE SciDAC program. SciDAC already has a modest data-management component. The SciDAC program partially addresses many issues relevant to data management, and has fostered close collaboration between computer science and application sciences. Serious consideration should be given to integrating the management of the new Office of Science Data-Management Program and that of SciDAC or the successor to SciDAC.

Introduction: Science in an Information-Dominated Age

We are entering an information-dominated age. Ability to tame a tidal wave of information will distinguish the most successful scientific, commercial, and national-security endeavors. Much elegant science has been performed over the centuries by subjecting simple observations to human intellect alone; but in the past few decades, our rising ability to automate observation and computation has opened otherwise inaccessible frontiers of the physical and biological sciences. The Office of Science has played a key role in these advances and has the ability and the responsibility to provide national and international leadership in information-intensive science.

Why should science face up to the tidal wave of information? Do we no longer believe in the search for elegant simplicity that has motivated scientists from Galileo and Newton to Crick and Watson? Simplicity of concept remains a guiding light in science, but all scientists know that wondrous complexity can arise from simple concepts. Our new information-enabled science allows us to dare to observe and model the complex—to describe the richness of all life based on a simple fourfold genetic code, to search for the bedrock of physical laws by measuring the immensity of the cosmos and the behavior of uncountable cosmic interactions recreated on Earth.

The scientific importance of managing data and information on an unprecedented scale is becoming clear—it is the limiting or the enabling factor for a wide range of sciences. At the most simplistic level, all sciences have needs to find, access, and store information. While the development of data-management technology is usually left to the computing industry, commercial efforts have been consistently inadequate to meet demanding scientific needs. As a result, many science programs have found themselves making mission-directed investments in data-management research, development, and deployment in order to meet their scientific goals. But, as the series of data-management workshops sponsored by the U.S. Department of Energy in 2004 made clear, such data-management efforts are inadequate and unbalanced.

Status of Scientific Data Management in the Office of Science

Currently, two-thirds of Office of Science research and development in data management lies within, and at the discretion of, the individual scientific programs. About $18M/year is spent by the programs on data-management research and development targeted at their most urgent needs. This is to be compared with the $9M/year spent on data management by DOE computer science. This highly mission-directed approach has been effective in meeting only the highest-priority needs of individual programs; it has not produced the coherent, leadership-class program of data management that will be essential to address the scales and nature of the Office of Science programs.

Not just research and development but also packaging and hardening as well as maintenance and support are required. Meeting these needs is a medium- to long-term effort requiring a well-planned program of evolving investment. Indeed, the larger program-centric data-management development projects are often started five or six years ahead of the required full-scale deployment.

An Office of Science Data-Management Program

To address this situation, we propose an Office of Science Data-Management Program at an initial scale of $32M/year of new funding.

The program should be managed by a Director charged with creating and maintaining a forward-looking approach to multiscience data-management challenges. The Director should strive to build a consensus across the application sciences on the scale and evolution of the budget for data management and on the evolving nature of the proposal solicitations that will define the program.

The program should favor collaborative proposals involving computer science and application science or, ideally, multiple application sciences. Proposals bringing substantial application science funding should be especially favored because such funding is a strong validation of the application science's urgent need. While collaboration should be welcomed, it will also be highly desirable that the collaborators can function as a single integrated interdisciplinary team whenever this approach is most appropriate. Involvement of the application sciences is expected to ensure that appropriate weight is given to hardening and packaging plus maintenance and support, in addition to relevant, career-enhancing computer science research. The Director should ensure that the proposal review process supports this approach.

The proposed program has many similarities to the DOE SciDAC program. SciDAC already has a modest data-management component. The SciDAC program as a whole partially addresses many data-management-relevant issues, while ensuring close collaboration between computer science and application sciences. Serious consideration should be given to integrating the management of the new Office of Science Data-Management Program and that of SciDAC or the successor to SciDAC.

Structure of This Report

Part I of this report presents the essential message: an overview of the science-driven requirements for data management and the recommendations resulting from the workshops.

Section 1 presents brief summaries of the science that is enabled by and challenged by data management.

Section 2 examines how the scientific investigation process involves storing, finding and accessing data and looks more specifically at the needs of the three groups of scientific activity: simulation-driven, experiment/observation-driven, and information-intensive. In the final part of this section, these needs are related to the detailed discussion in Part II.

Section 3 presents the recommendations arising from the workshops and summarizes the information on application-science priorities and on existing data-management investments that lie behind the recommendations.

Part II of this report systematically examines the data-management technologies relevant to science. A gap analysis shows where investment is needed.

Part I: The Essential Message

Science is the motivator for data management within the Office of Science. We therefore focus in this first part on eight representative scientific disciplines that are enabled by data management. As the brief summaries show, despite the differences among these diverse disciplines, they have striking similarities in their data-management needs.

We explore these needs by regrouping the eight applications into three categories: simulation-driven applications, observation/experiment-driven applications, and information-intensive applications. We define the concept of workflow, explore its role in the scientific investigation process, and examine the central workflow components in each of the application categories.

Based on this analysis, we identify six technology areas that are fundamental to supporting the data management requirements for scientific applications:

- Workflow, data flow, data transformation
- Metadata, data description, logical organization
- Efficient access and queries, data integration
- Distributed data management, data movement, networks
- Storage and caching
- Data analysis, visualization, and integrated environments

These six areas are discussed in depth in Part II.

We conclude Part I with a detailed recommendation for an Office of Science Data-Management Program. Forming the basis of our recommendation is information on application science priorities and on current data-management investments. We discuss not only the level of support needed but also a management approach designed to meet the evolving data-management needs of the science programs.

I-1 The Scientific Challenges

Dramatic improvements in scientific instruments as well as increasingly realistic simulation have resulted in enormous amounts of data and in concomitant challenges in managing that data. In this section we examine the data-management requirements of eight areas of science: (1) astronomy, astrophysics, and cosmology; (2) biology; (3) climate; (4) combustion; (5) fusion; (6) high-energy physics; (7) nuclear physics; and (8) nanotechnology.

I-1.1 Astronomy, Astrophysics, and Cosmology

We are entering a new era of precision in astrophysics and cosmology, driven on the one hand by an extraordinary array of new ground- and space-based observatories and the volumes of digitized information that they are supplying about our universe and on the other hand by large-scale and increasingly accurate simulations of the physical systems that give rise to the observable phenomena. As a consequence we are drawing new insights and making new discoveries about many fundamental questions regarding the nature of our universe, its contents, and its ultimate fate:

- How did the universe begin and how will it end?
- What is the nature of the dark matter and dark energy that appear to make up most of the universe?
- How do stars die, disseminate, and produce the elements necessary for life?

In order to shed light on these questions and others, new experiments are being planned that will probe the observable universe with unprecedented accuracy. The Large Synoptic Survey Telescope (LSST) [Tyson2002] will obtain repeat exposures of the entire night sky every two to three days, providing a dataset to search for transient objects such as supernovae with unparalleled efficiency and to measure the distortion in the shapes of distant galaxies by gravitational lensing. The SuperNova Acceleration Probe [Aldering2002], a proposed experiment for the DOE/NASA Joint Dark Energy Mission, will observe large numbers of supernovae at extremely large distances and will measure the change in dark energy properties over cosmological timescales. Accurate simulations of phenomena such as growth of structure in the universe and the explosions of supernovae will be essential to provide the theoretical framework for interpreting these observations and to allow the full precision of the data to be utilized.

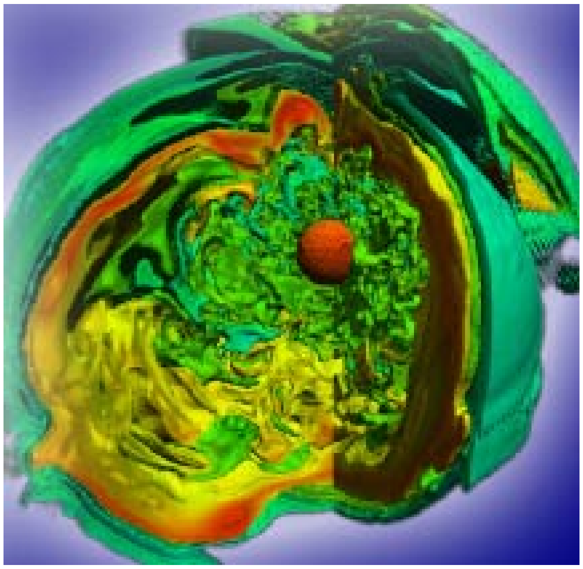


Figure I-1.1: Snapshot from a stellar explosion simulation [Blondin2003]. Capturing the complex, turbulent dynamics in a supernova environment is a challenge for computational astrophysicists and visualization experts alike.

Data management will be key to performing the ambitious programs outlined above. Three-dimensional simulations of stellar explosions being performed under the auspices of the DOE SciDAC TeraScale Supernova Initiative are currently producing data at the staggering rate of 5 TB per day, and the data aggregate produced will rise in the next few years from tens of terabytes to hundreds of terabytes per simulation. The LSST and other experiments will each produce up to 20 terabytes of data per night (see Figure I-1.1 and Figure I-1.2). These multiscale, multiphysics grand challenges are now being addressed, necessarily, by distributed, multidisciplinary teams. This trend will increase as data is accessed by communities encompassing thousands of users. In order to enable such collaborations, technology development is needed in data storage, networking, data analysis, data distribution, and visualization.

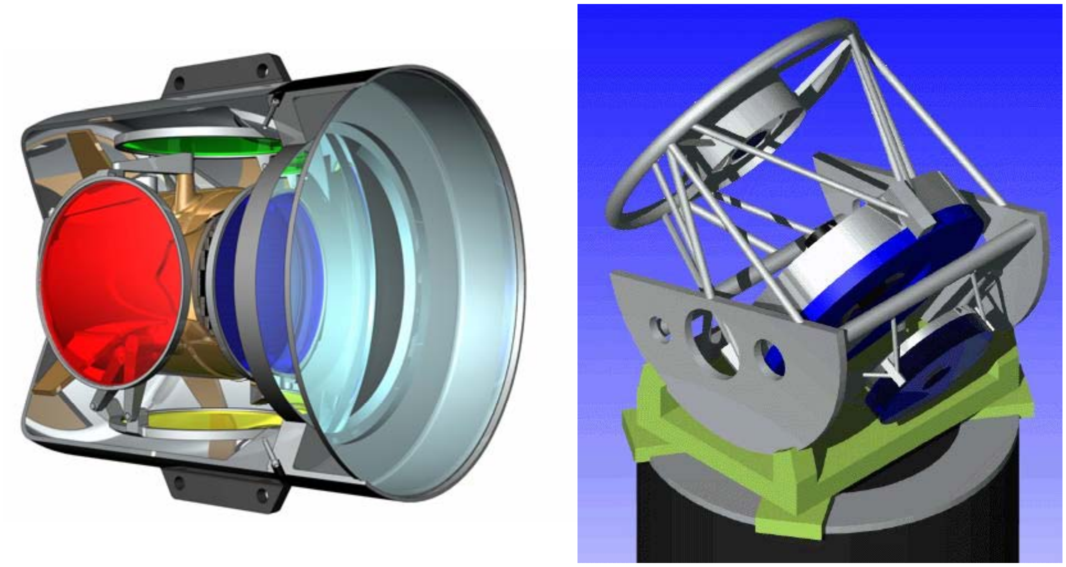


Figure I-1.2: Concept design for the proposed Large Synoptic Survey Telescope (LSST), which will record an image of the entire night sky every 2 to 3 days. The 3-gigapixel camera (left) will produce up to 20 terabytes of data per night.

I-1.2 Biology

Biological research is undergoing a transformation from a qualitative, descriptive science to a quantitative, predictive science as a result of the availability of high-throughput, data-intensive "omics" technologies, such as genomics, transcriptomics, proteomics, and metabolomics, together with the advance of high-performance computing. The generation and availability of community data repositories are revolutionizing the way biological research is conducted, creating a unique opportunity to apply a "systems" approach to address exciting new biological questions such as the following:

- What biochemical pathways control a plant's ability to create biomass or a microbe's ability to produce hydrogen?
- Can we identify natural populations of microbes that degrade or immobilize contaminants such as hydrocarbons or metals?
- What cellular repair mechanisms are employed by bacteria that live in environments of ionizing radiation?
- What communities of microbes are most effective in taking up excess carbon from the atmosphere?

High-throughput experiments and simulations already are generating vast amounts of complex data. For example, high-end Fourier transform ion cyclotron resonance (FTICR) mass spectrometers generate 20 GB per sample. High-throughput proteomics facilities such as those planned as part of the DOE Genomics:GTL program will be able to analyze hundreds of samples per day, providing hundreds of petabytes of data per year within the next decade. These data need to be analyzed, interpreted, and documented in order to create knowledge bases supporting meaningful comparisons of the results from one suite of analyses with another. Similarly, biomolecular simulations that relate structure and function of biological systems will be generating hundreds of gigabytes for each trajectory. All this information needs to be shared, annotated, archived, and made accessible to the general biological community.

The need for integrating the complex data types and derived information presents a fundamental challenge in data management because the data sources are large, diverse, and geographically distributed. New mechanisms will be needed throughout the data lifecycle to, for example, capture rich data and model descriptions; document data accuracy, quality, and uncertainty; integrate heterogeneous information from independent sources; and perform data mining and visualization of high-dimensional information. These data repositories and associated data-management services will provide a critical infrastructure supporting globally distributed teams of researchers developing models of cells, organs, organisms, and biological communities and using these models to improve our lives.

I-1.3 Climate

The Earth’s climate is produced by the nonlinear interaction of physical, chemical, and biological processes in the atmosphere, the world ocean, sea ice, and the land surface. These processes interact to maintain our current mild and hospitable climate. Nevertheless, over one hundred years ago, Arrehenius hypothesized that the climate would warm as a consequence of industrial carbon dioxide emissions to the atmosphere. The Office of Science has a mission to understand how energy production and use affect the environment, including the potential consequences of greenhouse gas warming. There is much about climate interactions that we still do not understand:

- How much internal variability exists in the climate system?
- What processes produce this variability?
- How will the climate system respond to changes in external forcing?
- Can we predict the evolution of the climate?

Climate system interactions cover a wide range of time and space scales, from a few hours and meters to many centuries and the entire globe. The datasets generated by both measurements and model simulations for analysis by climate researchers range in size from a few megabytes to tens of terabytes. Examples include raw measurements from satellite instruments, data from in situ observation networks such as the DOE Atmospheric Radiation Measurement program sites, and the output of three-dimensional global coupled climate models such as the Community Climate System Model (CCSM). Data from all these sources is maintained by several international institutions with varying levels of accessibility and technological sophistication.

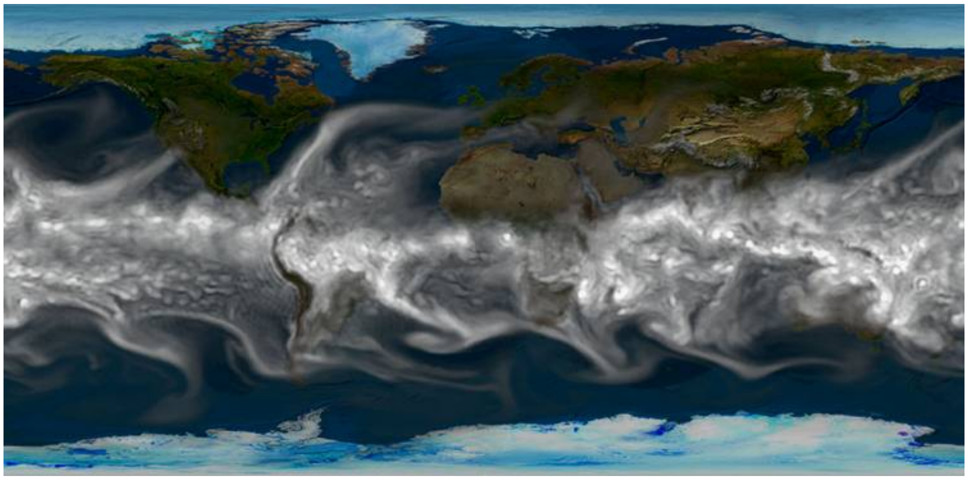


Figure I-1.3 High-resolution climate simulation using 70 km cells and generating 11 terabytes of data per 100-year run.

Many climate research studies use climate models to conduct simulated experiments on the climate system (see Figure I-1.3). For example, research groups in the United States and elsewhere are conducting climate change simulations with the latest versions of their climate models to provide results for the next report by the Intergovernmental Panel on Climate Change (IPCC). CCSM simulations for IPCC are being conducted at an unprecedented horizontal resolution for the atmosphere (180 km), and the early results are encouraging. Although the models themselves have benefited from computer science research, the tools that scientists use for data analysis have received less attention and can barely cope with the current data volume, such as the 7.5 TB produced by a single 100-year integration of CCSM. Already climate scientists spend half their time manipulating and analyzing data. In the near future, climate models will increase in resolution and will add algorithms to calculate the effects of unrepresented or underrepresented phenomena such as atmospheric chemistry and biogeochemistry. Satellite instruments scheduled for deployment will monitor a wider range of geophysical variables at higher resolutions, which will be used to validate climate models. All of these activities will overwhelm current capabilities and underscore the need for new technologies in data management and data analysis. The DOE SciDAC program has begun to address some of these issues with efforts such as the Earth System Grid, but more work must be done.

I-1.4 Combustion

Combustion science seeks to gain a predictive understanding of the combined effects of chemical reactions, multiphase fluid dynamics, and transport, which work together to release the chemical energy contained in fuels and oxidizers to generate heat and mechanical work. This science is important to improvements in fossil fuel combustion processes that represent over 85% of the energy used in the United States for transportation and stationary power generation. Finite fossil fuel reserves, environmental pollution, and climate change effects, as well as technological advances in materials processing, all drive the imperative for reacting flow science.

Using a strategy that layers data, models, and simulation and analysis tools, scientists are rapidly conquering the enormous range of physical scales and complexity in reacting flows to gain fundamental new understanding of important combustion processes. Researchers are just beginning to simulate laboratory-scale turbulent flames using massively parallel computers combined with emerging models and codes (see Figure I-1.4). These capabilities are enabling scientists to tackle long-standing fundamental questions that are key to gaining a predictive understanding:

- Can we learn new ways to control ultra-lean turbulent auto-ignition reactions to enable efficient, zero-emission engines?
- What fundamental changes and new possibilities are introduced by adding renewable hydrogen to combustors?
- How can we uncover the most compact chemical models and implement them adaptively in large-scale simulations?
- Can these and other validated submodels be developed to enable the science found at the larger scales of advanced experiments and real-world devices, or when the multiphysics complexities of complex fuels, soot, radiation, or sprays are introduced?

Such combustion grand challenges and the increasing value of large-scale simulations are placing significant data-management challenges in the path of combustion research. Whether we consider direct numerical simulations of turbulence-chemistry interactions or turbulence-modeling-based computations of device-scale combustion systems, there are significant challenges in managing the sheer volume of data as well as mining from it the intricate details that contribute new physical insights and models. Current computations generate about 3 TB of raw data per simulation, posing new data storage and movement challenges and requiring a new paradigm for data analysis. In many instances, regions of interest in turbulent combustion data are intermittent, both spatially and temporally, driving a need for automated, efficient, on-line feature detection and tracking algorithms tailored to detect relevant scalar and vector quantities. Adaptive steering and subsetting of data as it is computed are needed to enhance discovery and further analysis and visualization of events whose occurrence was not known a priori. Clearly, advances in data management are necessary to achieve the scientific progress promised by large-scale computational combustion science.

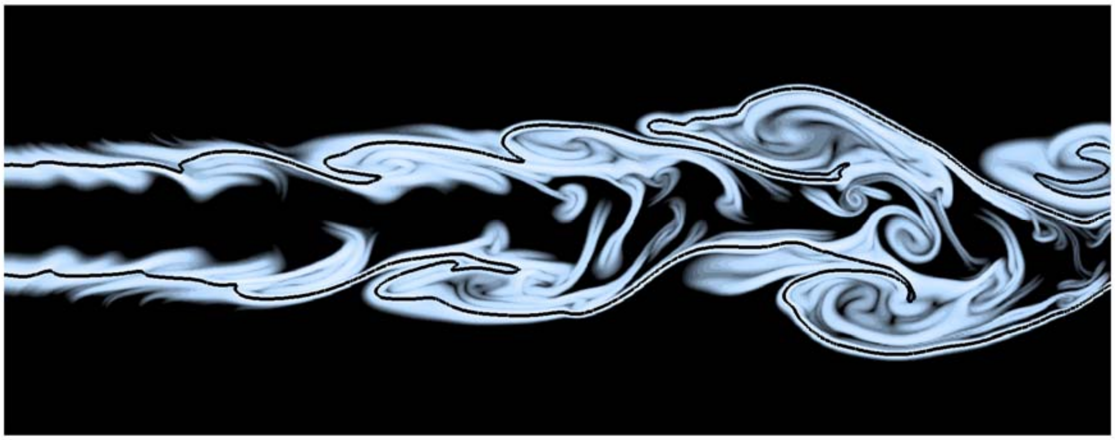


Figure I-1.4 Mixing in direct numerical simulation of a turbulent reacting CO/H_2/N_2 jet flame as revealed by the scalar dissipation rate isocontours. The black isoline corresponds to the stoichiometric mixture fraction.

I-1.5 Fusion

Plasmas constitute over 99% of the visible matter in the universe and are rich in complex, collective phenomena. A major component of research in this area is the quest for harnessing fusion energy, the power source of the sun and other stars, which occurs when forms of the lightest atom, hydrogen, combine to make helium in a very hot (~100 million degrees centigrade) ionized gas, or "plasma." A fusion power plant would produce no greenhouse gas emissions, use abundant and widely distributed sources of fuel, shut down easily, require no fissionable materials, operate in a continuous mode or intermittently to meet demand, and produce manageable radioactive waste. The development of such a secure and reliable energy system that is environmentally and economically sustainable is a truly formidable scientific and technological challenge facing the world in the twenty-first century.

The two principal approaches for confining the fusion fuel on earth are magnetic and inertial. Magnetic fusion relies on magnetic forces to confine the charged particles of the hot plasma fuel, while inertial fusion relies on intense lasers or particle beams to compress a pellet of fuel rapidly to the point where fusion occurs. In the past two decades, the worldwide programs have advanced our knowledge of magnetically confined plasmas to the point where we can confidently proceed to the larger-scale International Thermonuclear Experimental Reactor (ITER) burning-plasma experiment.

A number of grand challenge-scale plasma science questions need to be addressed in order to exploit the experimental program:

- What are the actual dynamics that govern the breaking and reconnection of magnetic field lines in a hot plasma medium?
- What is the best way to characterize plasma turbulence as a multibody problem involving collective dynamics?

- How can we unravel the mystery of the complex behavior of strongly nonlinear, nonequilibrium plasmas involved in atomic/material interactions with their external environments?
- When considered as an integrated system including the relevant physics on all important time scales, how will the global profiles of the plasma temperature, density, current, and magnetic field evolve over time?

In addition to dealing with vast ranges in space and time scales that can span over ten decades, the fusion-relevant problem involves extreme anisotropy, the interaction between large-scale fluidlike (macroscopic) physics and fine-scale kinetic (microscopic) physics, and the need to account for geometric detail. Increasingly realistic fusion simulations will result in large and diverse data demanding powerful data-management frameworks. In particular, ITER’s plasma production effort, planned around 2014, will generate an enormous amount of data, which will need to be collaboratively analyzed and managed in an international community.

I-1.6 High-Energy Physics

High-energy physics seeks to pose and answer the most fundamental questions about the particles and forces that make up our universe. High-energy physics, nuclear physics, astronomy, astrophysics, and cosmology are a set of deeply interconnected sciences bringing experiment, observation, theory, and simulation to bear on fundamental questions such as the following:

- Are there undiscovered principles of nature: new symmetries, new physical laws?
- Are there extra dimensions of space?
- Why are there so many kinds of particles?
- What happened to the antimatter after the Big Bang?
- How can we solve the mystery of dark energy and dark matter?
- What are neutrinos telling us?

Figure I-1.5 The Silicon Vertex Tracker is at the heart of the BABAR experiment at SLAC. Its millions of sensitive elements typify the evolution of detector technology that is making experimental high-energy physics ever more data intensive. (Photo courtesy of Peter Ginter)

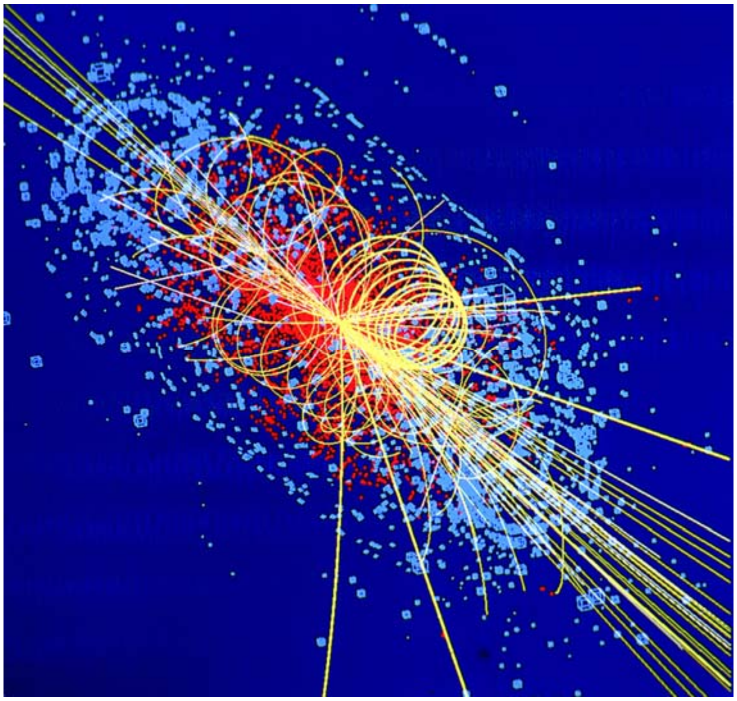


Figure I-1.6 Simulated decay of Higgs boson in the future CMS experiment at CERN. (Credit: CERN) Discovering and then studying the Higgs boson will require combing through petabytes of data.

It is an exciting time for experimental high-energy physics. Many of the questions may be answered in the next decade at the unprecedented collision energy of the Large Hadron Collider (LHC) at CERN,[1] complemented by matter-antimatter "factories" at the Stanford Linear Accelerator Center and KEK[2] and the current world's highest energy collider at Fermi National Accelerator Laboratory (see Figure I-1.5 and Figure I-1.6).

A common technical challenge runs through the past, present, and future of experimental high-energy physics: precision measurements in the quantum world of particles and forces require enormous statistics. Our ability to acquire and analyze huge volumes of data is one of the factors determining the feasibility and quality of the science. For example, collisions detected at the LHC will have a raw information content of close to a petabyte per second. Less than one-millionth of this information can be feasibly recorded and analyzed, so real-time decisions must be made by electronics and software to retain the millionth thought likely to reveal new physics. Even with this selectivity, physicists must seek revolutionary new approaches to data management and data analysis to allow scientific intuition and intellect to range unhindered over a daunting volume of data .

I-1.7 Nuclear Physics

Taking a step up from pondering the most fundamental particles and forces in nature, we can begin to ask questions about how the fundamental particles are brought together to form complex massive particles and what characterizes the forces that bind these particles. Here we are entering the realm of nuclear physics.

- How is a proton constructed from three quarks and a field of gluons? We know that quarks account for 2% of the mass and 25% of the spin. How does the internal structure of protons and neutrons give rise to the binding and properties of the thousands of nuclear isotopes we find in nature?
- In the Big Bang model of the universe, hadrons (protons, neutrons, mesons) formed during a period of expansion and cooling when the universe was about a microsecond old. What are the properties of the primordial plasma of quarks and gluons before the phase change to hadronic matter?

Extensive programs in experimental and theoretical nuclear physics are making progress toward answering these questions; but as in all science, new insights give rise to new questions. The experimental programs have ever-increasing datasets; some investigations focus on a single, large data sample whereas others analyze the correlations across data samples. The scale of the data-handling issues is characterized by experiments having peak data generation rates of tens of megabytes per second, the major programs generating of order one petabyte per year, and data analysis environments having tens to hundreds of scientists simultaneously accessing refined datasets of tens of terabytes (see

[1] CERN: European Laboratory for Particle Physics, Geneva, Switzerland. The CERN LHC program involves major U.S. participation.

[2] KEK: High Energy Accelerator Research Organization, Tskuba, Japan.

Figure I-1.7). Elements of the computational theoretical nuclear physics programs have similar characteristics to other simulation sciences, with significant needs for high-performance parallel I/O attached to massively parallel computers, as well as geographically distributed data flow for small teams of scientists to share and manipulate data on the appropriate facilities.

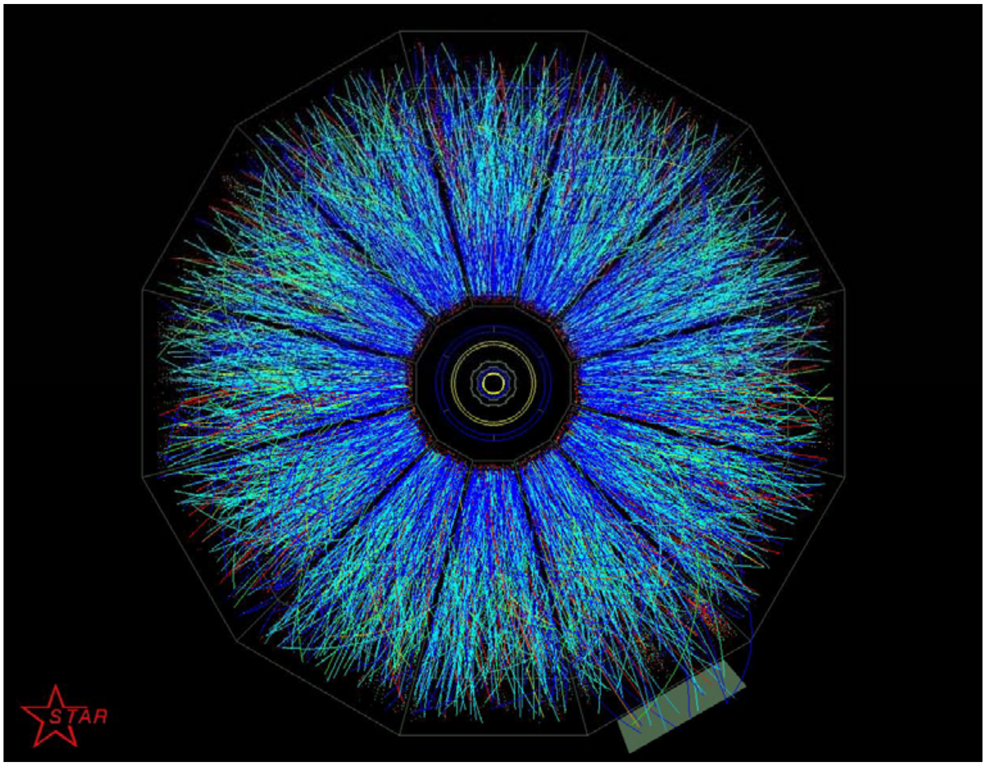


Figure I-1.7 Gold-gold nucleus collision measured by the STAR detector at the Relativistic Heavy Ion Collider. The STAR detector can produce 2 gigabytes/s of compressed data.

I-1.8 Nanotechnology

As the needs of our high-technology society have advanced, so have our demands for new materials that are stronger, lighter, and cheaper yet perform well under severe conditions. Nanoscale features and molecular assemblies can have properties that are dramatically different from traditional materials, surfaces, and catalysts, offering enormous potential for meeting some of these pressing demands.

Researchers in nanophase materials uses diverse instruments and techniques, including electron microscopy, X-ray diffraction, neutron scattering, and nuclear magnetic resonance. The new DOE nanoscience centers are being placed near major microscopy, synchrotron, or neutron-scattering user facilities to support this research.

New facilities and instrumentation such as that built at the Spallation Neutron Source (SNS) at Oak Ridge National Laboratory provide orders of magnitude more neutron flux and larger detector arrays than predecessor facilities, with concomitant increase in data volume. At full capacity, SNS expects to have 24 instruments and plans to accommodate

1,000 or more guest researchers per year. Likewise, next-generation electron microscopes will be capable of taking much more detailed (and larger) images at shorter time intervals, as well as spatially resolved spectra, which increase data output by orders of magnitude.

Nanoscience is young and is not yet straining against the limitations of the science of data management. However, data management is already a challenge, and there are growing needs to handle both complex and high-volume data that will be well served by exploiting developments driven by the other sciences.

I-2 The Roles of Data in Science

Many scientists feel challenged by the quantity and complexity of their data. To paraphrase the comments of many workshop participants, “I’m spending nearly all my time, finding, processing, organizing, and moving data—and it’s going to get much worse.” The first hurdle faced at the workshops was to turn this sense of dread into a well-organized statement of technological needs. Application scientists (researchers in the science domains of the previous section) rapidly discovered that science could be grouped into three types of activity with similar problems: simulation-driven science, experiment/observation-driven science, and information-intensive science. These scientists, armed with the certainty that they were not strange outliers, even if their ability to speak computer science jargon was limited, were able to explore how their data-management problems related to topics that made sense to the computer scientists.

In this section, we examine the application-science needs using the three groupings that arose at the workshops. We then outline how computer scientists see the problems; a detailed examination of the issues and gaps from a computer-science viewpoint is presented in Part II.

Before looking at the three-way grouping of application-science needs, we briefly examine the data flows and workflows used by scientists.

I-2.1 Data Flows and Workflows in Science

The workshop participants considered both spiral and linear models[3] as ways of unifying the description of how science is done and how information flows. The spiral model describes well how a series of exploratory and confirmatory investigations lead to a growth of knowledge, but it is a poor vehicle for understanding the data flows in a single investigation. A simple, almost generic example of the linear model is shown in Figure I-2.1.

[3] Software developers debate the merits of describing the software creation process with a linear model (perceived need leading to shrink-wrapped product) or a spiral model (it’s never finished: the existing product just help researchers understand the needs for the new, improved product). Scientists spend (perhaps) less time thinking about a good model for the scientific process.

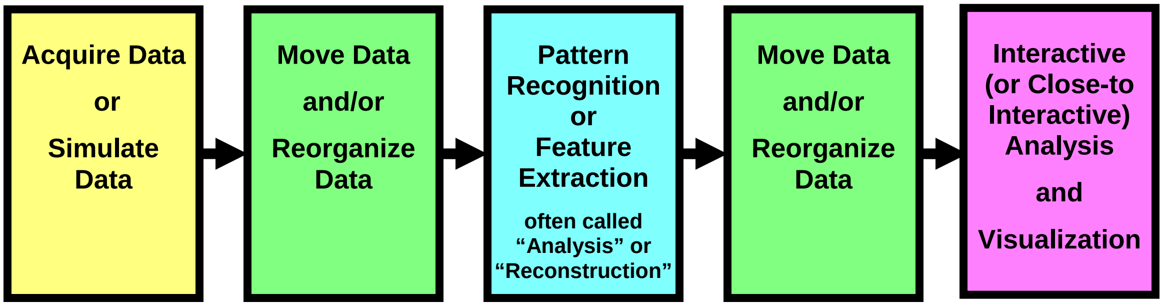


Figure I-2.1: Simple view of a data flow and workflow in a scientific investigation.

Data often must be moved because the national facilities used to acquire or simulate the data are separate from the analysis facilities available to scientists. In more complex collaborative activities, data may even be moved to national centers in other countries to perform resource-intensive processing.

Data frequently must be reorganized, for example to collect the subset of the data that one group of scientists intends to study. Reorganizing a gigabyte of data can take a few minutes on a workstation; reorganizing a petabyte can take months, monopolizing hardware worth millions of dollars.

Pattern recognition and feature extraction are the keys to taming datasets too large to study directly. In many cases they are simply an automation of the visual searches for patterns and features that can be done by eye on small datasets. However, once the patterns and features have been extracted and stored in a more compact dataset, their analysis presents completely new challenges.

For those scientists still working in a mode where the acquired or simulated data can be directly visualized, Figure I-2.1 collapses to just two boxes. However, this mode is becoming rare. Indeed, the three boxes in the middle of the figure occupy more and more of application scientists' time. The central box—pattern recognition and feature extraction—at least has some intellectual content relevant to the science, but the data-movement and organization activities reflected in the other two boxes are becoming increasingly onerous.

A framework automating these activities would vastly enhance scientific productivity, particularly in data-intensive science conducted by small teams. Such a framework would also automate the capture (and audit) of all the steps taken by all participants so that the data provenance was assured. Such assurance becomes vital as small teams evolve into larger teams and then into worldwide collaborating communities.

Figure I-2.1 hides the hardware and software components that accomplish the actions. Figure I-2.2 illustrates some of the hidden components that accomplish the multiple data-related actions performed in many experiments and simulations. The top layer illustrates the control activities, the middle layer the software components, and the bottom layer the physical resources needed for the activities.

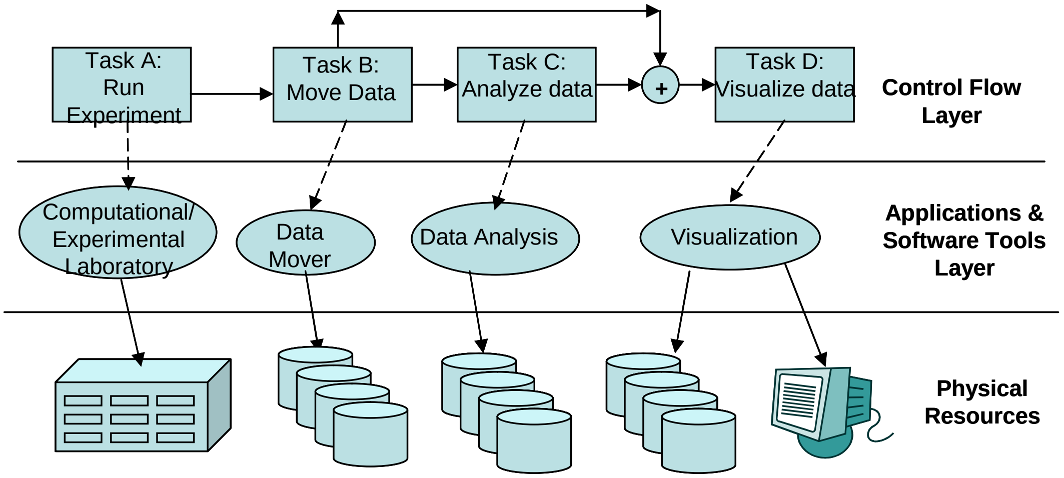


Figure I-2.2: Example of a workflow created in the scientific investigation process, showing the three layers: control flow, applications and software tools, and physical computer hardware.

I-2.2 Simulation-Driven Applications

Many simulation scientists collaborate in small groups in most stages of the scientific process. Increasingly, however, scientifically important problems require large, multidisciplinary teams. In these instances, the need to access distributed data and resources is the rule rather than the exception. Scientific discovery requires that we ultimately create distributed environments that not only facilitate access to data but also actively foster collaboration between geographically distributed researchers.

Typically, simulations are executed in batch because they are long running and the computational resources are located in a few supercomputing centers. Increasingly, however, simulation scientists are expressing the desire for interactive capabilities that will enable data management, analysis, and visualization “on the fly.”

Regardless of the simulation domain or execution mode, the sizes of generated data are very large. For example, three-dimensional hydrodynamics simulations performed by the DOE SciDAC TeraScale Supernova Initiative are currently producing data at the rate of 5 TB per day. More detailed and higher-dimensional simulations required for predictive science will drive data rates upward at an exponential rate. If the growing data monster cannot be tamed, hopes for scientific progress will be dashed. Major efforts are needed to ensure that scientists are provided the data-management tools required for innovative scientific investigations.

While the particular steps performed by simulation scientists to obtain and analyze scientific data may differ significantly, three categories emerge as the central workflow components of simulation-driven science: data movement and reorganization, data analysis, and visualization. All involve data-management challenges.

Data Movement and Reorganization. Simulated data are often written out as thousands of files, in order to allow the supercomputer to perform I/O without bottlenecks. Hence, there arises a need for significant parallel I/O development and support. This begins with the need to define a portable, efficient industry standard and includes the need for interoperability between parallel and nonparallel I/O. Scientists must also store large, distributed datasets. While archival storage will be required, a significant fraction of the simulation data must be postprocessed and analyzed *as it is produced,* which in turn will require the ability to cache data on this scale. The processed data also must be augmented by metadata and annotations tracking their provenance. (Provenance may include information on the version of the code used to perform the simulation, parameters for both the simulation itself and the models, information on simulation input, the machine configuration used when the simulation was performed, and information about the compilers used.) In addition, researchers must be able to transfer the data efficiently; a potentially integral part of data transfer in a distributed context is data compression.

Data Analysis. As volumes of simulated data increase, scientific discovery by visually rendering raw simulation data becomes impractical. Derived quantities often lend themselves best to scientific discovery. Data analysis prior to visualization may require data transformation; feature detection, extraction, and tracking; inverse feature tracking (clustering and correlation); and statistical analysis. For example, data may be mined from many files in order to identify and then track regions containing particular types of information, such as flame fronts. Data analysis also should be coupled with visualization. Moreover, there is a clear need for parallel data analysis routines that can be coupled with simulations run on today's—and tomorrow's—advanced computer architectures.

Visualization. A principal role of visualization is the extraction of scientific understanding from the tractable datasets emerging from analysis. Visualization is also required to instrument intermediate stages of the computational pipeline, for example to see whether unexpected output from the simulation is confusing feature-extraction code. Long-running simulations can become vastly more productive if some information can be visualized in real time, allowing decisions to abort or steer the simulation. Latency can be critical in these applications. Visualization routines should be able to understand the common data model defined in the data workflow so that simulation scientists can easily create new visualization networks for specific application domains.

I-2.3 Observation/Experiment-Driven Applications

As with simulation applications, experimental and observational applications are dealing with ever-increasing data volumes, some of which will reach petabytes per year within the next few years. The challenges in managing these large datasets are driven by the diversity of requirements for the storage, organization, access to, and curation of data at different stages of the workflow process.

I-2.3.1 The Workflow

In the *data acquisition* phase of an experiment, data is collected by digitizing detectors and stored in a raw instrumental format. Data rates can be high enough that simply recording the data in real time can be a challenge. In some experiments (e.g., in high-energy physics), rates are rapidly approaching petabytes per second, well beyond those that can be stored and retrieved by today's technology. Hence, real-time processing is done to determine which elements of data are likely to be interesting. These are recorded, while the vast majority of data is simply dropped.

In the *data-processing* phase, data is transformed from instrumental format to a form that has some scientific meaning and has identified the important features in the data (e.g., raw events are processed into electron trajectories and energies). If the experiment is long running, this stage can be stable and repetitive, well suited to automation and coarse-grained parallelization. In some experiments the raw data is then discarded as being too large to save in any practical manner. Data processing can be complex: data subsets may have complex interrelations, necessitating one or several intermediate persistent datasets. Generic workflow tools must be flexible so they can be tailored to each experiment's specific needs.

In the *data analysis* phase, the data is accessed by large scientific communities spread across multiple institutions. Data analysis can involve extensive visualization, complex queries, and cross-correlations among different datasets. By its very nature, this phase is dynamic and unpredictable. In some fields (e.g., astrophysics and biology), datasets from one experiment are analyzed in conjunction with datasets from other experiments. In other fields (e.g., fusion), datasets are compared with predictions from simulations. It is often desired to replicate datasets in multiple locations and reorganize them for more efficient analysis, but the sheer size of the datasets can make replication or reorganization take months.

I-2.3.2 Technical Challenges

Several key technical challenges are shared by current and future experiments.

Storage. The low-level technologies for constructing large storage systems are being stressed. Moore's law does not apply equally to all aspects of storage systems. Storage capacity is growing faster than bandwidth and access times, so we are driven to constructing massively parallel I/O systems to maintain throughput. Some experiments (e.g., in high-energy physics) necessarily access large numbers of kilobyte-sized chunks of data, which is an access pattern poorly matched to existing storage technologies. Equal ease of access to all bits of a large dataset is often not necessary. Data-caching techniques can be valuable to provide high-speed access to interesting subsets of the full dataset. Data integrity is important. Hardware and networks are not perfect, so data loss and corruption must be caught and fixed. As systems grow in size and complexity, problems may pass unnoticed until recovery becomes difficult and expensive.

Data organization. A problem shared with simulations is data organization. Multiple processing versions exacerbate the problem. Data is seldom organized optimally for access during the analysis stage (e.g., by position on the sky). Instead, it typically is organized in the time order collected. Data reorganization can sometimes be cast as a data query (e.g., fetch all objects that match some search conditions). Relational databases (with indexing) provide much of the needed functionality, but they are currently unable to handle petabyte-scale datasets; further, the relational model is often poorly matched to the complex relationships needed in a database of processed experimental data. Some specialized data-access tools work on files (e.g., ROOT), but generic solutions of this type do not yet exist.

Data analysis. Large, data-intensive experiments can involve over a thousand scientists at hundreds of institutions in several countries. Data analysis is a major scientific challenge in itself and motivates nations to seek create their own centers of excellence that are funded by different sources from the main experiment. Grid technology to allow these distributed resources to be integrated is vital for scientific success.

Data provenance. In experiments with hundreds of scientists and thousands of raw and processed data products, keeping track of data provenance is of high importance. Metadata and data model standards become even more important to ensure that datasets can be readily understood by users from outside a particular experiment.

Data archiving. Experimental data generally have archival value: there are many examples of new understanding being extracted from data over ten years old. The archiving of data places demands on having well-defined metadata, robust storage, and open access mechanisms to the data.

I-2.4 Information-Intensive Applications

In some research areas, most notably biology and more recently homeland security, increases in computing, network, and data-storage capabilities are revolutionizing our ability to understand systems-level phenomena. In other areas, particularly combustion chemistry and nanoscience, such a systems-level approach is beginning to enable researchers to model dependencies between phenomena at scales from atoms to devices, directly connecting basic research with engineering application.

The data-management challenge for systems-oriented research is not simply about data volume. More critical is the fact that the data involved is produced by multiple techniques, at multiple locations, in different formats and then analyzed under differing assumptions and according to different theoretical models. The need to understand such a heterogeneous collection of information involving thousands to billions of individual datasets, at the scale of communities and across disciplines, defines the core challenge faced in information-intensive applications. In essence, systems-oriented research aims to produce “big science” results by integrating the effort of thousands of independent research programs.

To understand some of the issues facing information-intensive applications, consider a biology example involving measurements of the concentration of thousands of proteins in a cell as a function of exposure to a chemical contaminant. The protein concentrations can depend not just on the contaminant but on many factors such as the genetic sequence of the cells used, growth conditions, and cell age. In order to federate data from multiple experiments, all this metadata must be made explicit and persistently associated with the data. Further, assume that the experimental data will be compared with a simulation that accounts for the three-dimensional distribution of proteins within cells. The experimental data, which has no spatial information, must now be combined with additional information (e.g., microscopy data and knowledge about which proteins are usually found in various cell regions) and translated into the data model and format expected by the simulation. Conclusions about whether the simulation model accurately represents the cell depend on the entire chain of data and the assumptions made at various stages. As research progresses, scientists might wish to automatically scan community data resources and re-evaluate the model as cells are studied under new conditions, more information about protein distributions is discovered, the model itself evolves, or new experimental techniques (with their own data models and formats) are developed and provide additional types of information about the cell.

As the example illustrates, the complexity involved in information-intensive research is tremendous, and the ability to track metadata and relationships is quickly becoming a limiting factor. Scaling these capabilities to the community level—which involves capturing additional information, publishing data and metadata, curating public data resources, enabling documentation of relationships between different types of data maintained by different subcommunities, and providing reliable data and metadata discovery and access services to potentially billions of datasets to tens of thousands of researchers—is truly a grand challenge. Automation of this process, with robust tools that allow researchers to easily configure and control the underlying work, data, and metadata flows, will be a critical factor in realizing the promise of informatics-oriented research.

Data-management tools must transparently support scientific research processes. In the same way that researchers performing data-intensive experiments and simulations should not need to become data managers to pursue their goals, those working in information-intensive domains should not need to become knowledge engineers. Simple standard ways of viewing metadata and discovering data based on queries about its metadata and relationship to other information, a minimally invasive infrastructure to capture required metadata, and mechanisms to rapidly create, evolve, and map between semantic descriptions of data and of data processes will all be required. These capabilities may in turn need to be aggregated into coherent, knowledge-aware suites of experiment planning, design, and execution tools.

Information-intensive techniques have already proven their value in areas such as bioinformatics, and they promise to fuel the next generation of research and development across many domains. The requirements noted here represent significant challenges in data management. However, the existing knowledge and technology base across data and information management, distributed computing, and semantic information processing strongly suggest that the requirements can be met. Success in this area will allow researchers to tackle complex, high-priority issues with an unprecedented breadth of expertise and resources.

I-2.5 Foundation of Scientific Data-Management Technology

The needs described above cover many aspects of data-management technologies. We organize the needed technologies into six areas, discussed briefly below and in detail in Part II:

I-2.5.1 Workflow, Data Flow, Data Transformation

Workflow management is a unifying need in virtually all science areas. The specification of scientific workflows is not simple, however, since it covers the tasks to be performed and the flow control specification, the software components to be used, the data flow requirements between the components, and the storage systems involved. Workflows need to explicitly express the synchronization between tasks and to identify whether the tasks are used repetitively. In addition, there is a need to specify explicitly the data transformation tasks that must be performed in order to have the output of one component formatted properly as input for the next component.

I-2.5.2 Metadata, Data Description, Logical Organization

Metadata refers to the information on the meaning of the data being generated and collected by scientific applications. Metadata is essential for scientific investigations: without the orderly collection of the metadata, the data is of no value. The structure of the data—the data model—is also essential information. Some file systems, such as netCDF, have a header that indicates the structure associated with each file, but this is not sufficient. Additional information is needed, such as the units used, the person who generated the data, and the significance of the results. Another important aspect of metadata is the history of how data was generated—its provenance. Ongoing community annotation of data with additional notes about its quality and its relationships to other data is also becoming a key capability.

Automating the collection of metadata becomes a necessity at the scales being discussed, although some metadata, such as the unstructured information traditionally kept in notebooks, may continue to be entered manually, with quality and completeness managed by policies and procedures. Moreover, the semantics (terms, meanings, and relationships among terms) of data and metadata models also needs to be as precise and as standardized as possible to support data interpretation and integration. Full descriptions—the “ontologies”—can be powerful: their structure, such as broader terms and narrower terms forming hierarchies, may be sufficient to automate many aspects of data integration. To assure that scientific data retains its meaning and value as it is shared among researchers and over time, scientists must have access to flexible, easy-to-use metadata technologies.

I-2.5.3 Efficient Access and Queries, Data Integration

By efficient access we mean the ability to write data into a storage system and retrieve it efficiently. A consequence of dealing with large quantities of data is the need to find the subset of the data that is of interest. Often that means searching over billions of data objects, using several descriptors (attributes, properties) for the search. Searching can be facilitated by efficient high-dimensional indexing methods. Much of the scientific data is stored in files, with specialized formats. There is a need to provide data-querying capabilities over the content of such files, such as a general-purpose query system, similar to a database-management system but allowing the data to stay in the scientists’ files. This appears to be a unique requirement by scientific applications that is currently not addressed by the database-management industry. Another aspect of accessing data is the need to integrate data from multiple sources, perhaps in multiple formats and data structures. This is common for applications that correlate interrelated aspects of a system, such as biology applications in which genomics, proteomics, microarray, and spectral data must be correlated and integrated.

I-2.5.4 Distributed Data Management, Data Movement, Networks

Attempting to move large volumes of scientific data exposes many bottlenecks. Even within a single site, the rates at which data can move between workflow components may be a constraint. When data is moved over wide-area networks, the difficulty is not only in having sufficient bandwidth but also in dealing with transient errors in the networks and the source and destination storage systems. Thus, moving a terabyte of data becomes a major task for the scientist. Grid middleware technology can be helpful, especially middleware components that perform monitoring and recovery for transient failures. A technique for avoiding repetitive data movement is replication of selected subsets of the data in multiple sites. Replication requires placement strategies based on actual and projected usage. Data can be placed not only in computer-center storage systems but also within a network fabric enhanced with temporary storage. Grid technology is already beginning to address such issues, by providing Grid storage management, data-movement tools, and replica catalogs.

Management of user authentication and authorization to read or modify the data is vital. Even in a totally open environment, it is a disaster if one scientist’s mistake silently corrupts the data that a thousand colleagues are studying. Clearly a data-security infrastructure is needed that makes it easy to apply security while minimally burdening the scientist.

I-2.5.5 Storage and Caching

Reliable, robust storage technology is essential for scientific data. Some scientific data, such as experimental or natural phenomena observations, is irreplaceable, and thus scientific investigation cannot tolerate undetected data-retrieval errors. In several Office of Science programs, the disk, tape, and server technology for data storage already dominates computing costs. It is important that the scientific community continue to work with storage system vendors to ensure the availability of affordable, reliable storage systems.[4]

Storage hardware must be used effectively. Large-scale simulations can produce data at a rate much faster than a single storage system can absorb it. Similarly, instrument data can be generated a very high rate and needs to be moved to storage systems at that rate. The obvious solution is parallel I/O, but without adding complexity for the scientist, especially when data must be moved across the country or between computers.

Another issue involving the efficient use of storage systems is the management of files that are staged from robotic tape systems. When a large volume of datasets is generated, the data is typically archived to tape, but only a portion of the data (areas of interest) needs to be moved to disk. The technologies for automating the process of moving needed data from tape to disk (while making the migration imperceptible to the

[4] For example, the high-capacity tape cartridges currently used by the most data-intensive scientific programs were developed as a result of interactions with a leading vendor.

application) and for choosing what to keep in cache are important capabilities for scientific data management.

I-2.5.6 Data Analysis, Visualization, and Integrated Environments

Scientific investigation requires various tools for data analysis and visualization, as well as integrated environments in which these tools can operate seamlessly. As the size and complexity of the raw data generated by simulations, experiments, and observations increase, researchers will increasingly rely on analysis and visualization techniques to preprocess and summarize the data into a more comprehensible form. The derived data and visualization output then become the primary results that are shared with the community. In order to generate this derived data, advances in data analysis techniques are needed, including improved feature identification and tracking, sophisticated representation and search algorithms for find regions similar to a query region in a database, real-time anomaly detection in streaming data, and scalable algorithms that can operate on different types of raw data.

Further, with many different modalities of data coming on line, such as Web documents, experimental data, and journal papers, researchers are also interested in mining such data to find interesting associations. Visualization tools must be able to handle multidimensional datasets and scalable algorithms. New approaches for comparative visualization and 3-D data exploration need to be developed to aid the scientific investigation process.

It also is important that these analysis and visualization tools be available, not only as standalone modules but also as part of an integrated environment where a researcher can easily work with different tools, without having to spend a lot of time on cumbersome and computationally expensive data transformations. In addition, uniform data formats are required to support different computer environments ranging from desktops to large supercomputers.

I-3 Recommendation: A Scientific Data-Management Program for the Office of Science

The data-management workshops implicitly posed and explicitly answered the following questions:

- Is data management a critical-path problem for DOE science domains?
- Are there major unmet, or inadequately met, needs?
- Is there substantial commonality between the sciences in data-management problems?
- Can computer scientists and application scientists work together to address these problems?

Five years ago[5] the path forward was unclear. Today, as these workshops demonstrated in their presentations and even more in the working discussions, the answer to all four questions is a resounding "yes."

These findings argue strongly for an Office of Science Data-Management Program that will provide the needed capabilities for DOE's scientific challenges by coordinating existing research and development efforts and bringing to bear additional resources that achieve a long-term vision on the scale warranted by the science mission.

The workshops could capture only a snapshot of the current perceived needs for progress in data management. Since data-management needs evolve even faster than data-management technology, the Data-Management Program must incorporate ongoing strategies for determining and applying priorities across the Office of Science. Here, we review the current Office of Science data-management efforts, identify the needs and priorities for additional research and development in data management, and present an approach for effectively carrying out such a program.

I-3.1 Existing Office of Science Data-Management Effort

Table I-3.1 summarizes the current Office of Science data-management effort. Many of the numbers are not precise and auditable, but they do represent the best estimates of involved scientists. All the efforts in this table are restricted to data-management research, development, deployment, hardening, and maintenance, excluding operations and equipment. For projects that are not uniquely focused on data management, an attempt has been made to estimate the portion of the project effort that is devoted to data management. In the case of the application sciences listed in the lower part of the table, the resources devoted to data management were estimated by the scientists who

[5] For example, in October 1998, an ad hoc DOE Data Management Workshop was held at SLAC. Much valuable information was exchanged, but no clear need for action emerged from the workshop. Workshop presentations are at http://www-user.slac.stanford.edu/rmount/dm-workshop-98.

participated in the workshops. How complete is the picture? Only Office of Science funding is shown. As the table shows, the existing level of ASCR support for Office of Science data management is small compared with the investment that the application sciences feel is needed.

Table I-3.1 Existing Office of Science projects or activities with data-management components. See text for explanation of columns.

	Resources Expended by DOE for Data Management Activities in $M/yr	
Project or Activity	**Computer Science Efforts**	**Application Science Efforts**
SciDAC: Scientific Data Management ISIC	3.0	
SciDAC: Particle Physics Data Grid	0.5	0.6
SciDAC: High-Performance Data Grid Toolkit	0.8	
SciDAC: DOE Science Grid[6]	0.2	
SciDAC: Fusion Collaboratory	0.4	
SciDAC: Earth System Grid II	1.8	0.4
SciDAC: Logistical Networking	0.3	
Collaboratory for Multi-Scale Chemical Science	1.2	
Storage Resource Management for Data Grid Applications	0.5	
Scientific Annotation Middleware	0.6	
Astronomy and Astrophysics		0.6
Biology		2.4
Climate		4.0
Chemistry/Combustion		0.1
Fusion		4.0
High Energy Physics		5.0
Nuclear Physics		1.0
Nanoscience		0.1
TOTAL Existing Activity	**9.3**	**18.2**

[6] Terminated August 2004.

I-3.2 Needs and Priorities

As a result of interactions with computer scientists at the workshops, application scientists were able to reach a clearer understanding of the areas of computer science and technology that were relevant to their current and immediate future problems. These needs and priorities are summarized in this section.

I-3.2.1 Overall Priorities

During the final workshop the application scientists were asked to make their best estimates of their priority ranking for the major areas described in Section I-2.5. Sciences were allowed to consider themselves simulation-intensive and/or experiment/observation-intensive and/or information intensive. The results are shown in Figure I-3.1.

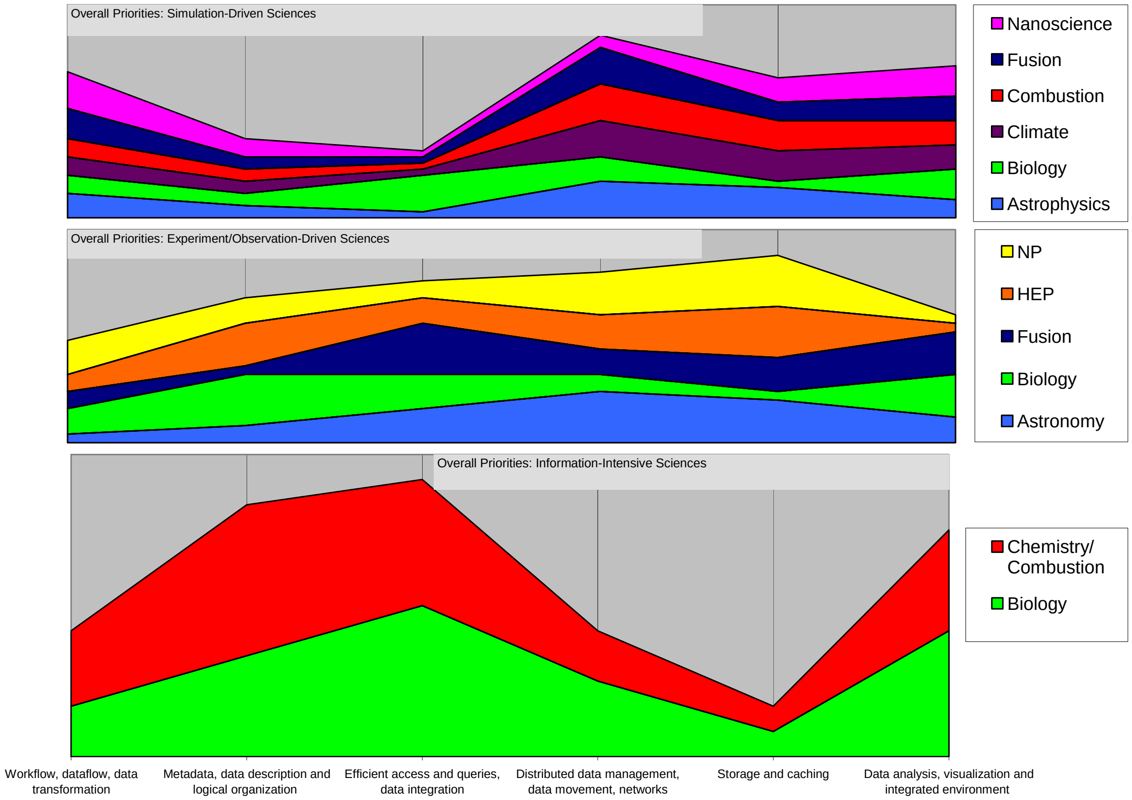


Figure I-3.1: Overall priorities for each of the six areas of data management outlined in Section I-2.5 and discussed in detail in Part II. Each branch (simulation-driven, experiment/observation-driven, information-intensive) of each application science ranked the six areas from 1 (lowest) to 6 (highest).

The priority assignments show many strong similarities among the sciences in each of the three categories. Even where differences exist, they may be more in timing. For example, traditional scientific visualization, focused primarily on visualization of continuum fields, has had little application in high-energy and nuclear physics; however, we can anticipate a greater role of more advanced visualization techniques in the future.

I-3.2.2 Priorities for Additional Effort

A complementary and even more probing request was made to each application science in an attempt to discover the urgent priorities for applying additional effort: “Imagine that your science has obtained funding for four FTEs to work on data management. Where would you put them to work?” Half-FTE assignments were allowed, and sciences were encouraged to consider the possibility of pooled effort in areas of common need. The results are shown in Figure I-3.2.

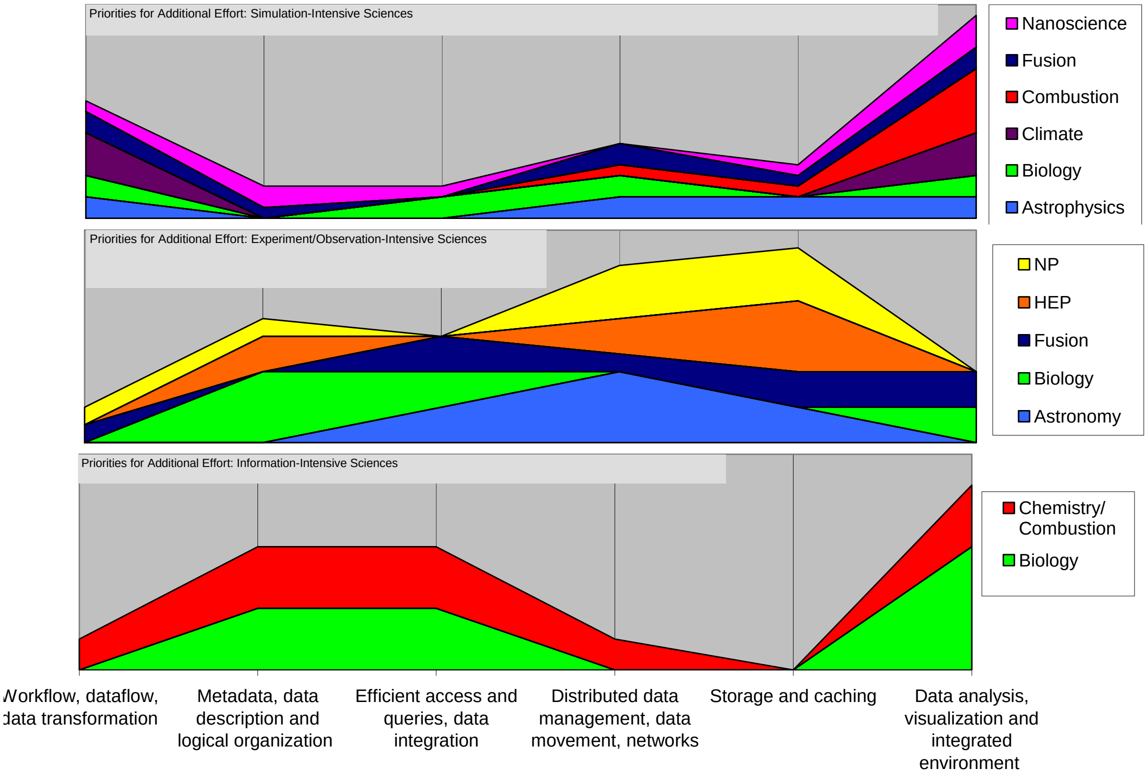


Figure I-3.2: Priorities for additional effort for each of the six areas of data management outlined in Section I-2.5. Each branch (simulation-driven, experiment/observation-driven, information-intensive) of each science imagined how they would assign a very limited amount of additional effort.

In several cases, sciences declared areas “high overall priority” but assigned few or no FTEs, explaining that because these data-management needs were already being addressed in some way, the most urgent investment must be elsewhere. For example, the astronomers did not assign additional effort to metadata issues because of the large (over $10M) international effort currently devoted to astronomy catalogs.

The very restricted amount of additional effort that the sciences were allowed to allocate highlights stark differences between simulation-driven, experiment/observation-driven, and information-intensive sciences. However, summing over all types of application sciences, every area of data management requires significant additional effort.

I-3.2.3 The Problem with the Status Quo

The workshops clearly demonstrated that the Office of Science programs have growing needs for data-management science and technology and that the needs of the programs have much in common. The current approach, with a few honorable exceptions, is "leave it to the science programs to fund their own data management." This approach does too little to address the looming technology gaps and fails to exploit commonality in the needs of the programs.

An additional major issue, appearing again and again during workshop discussions, was the difficulty of funding the hardening and packaging and the deployment and maintenance of the good solutions that arise from DOE research. The result has often been that first-class computer science funded by ASCR has been unusable by the application sciences because there was no means to put computer science results into practice. Opportunities for U.S. science to capitalize on revolutionary data-management developments are being lost.

I-3.3 Setting the Scale of a Data-Management Program

The scale of the additional resources was estimated from both the computer science and the application science perspectives:

- The computer science participants were asked to estimate a minimum level of computer-science effort required to make appropriate progress on each of the subtopics appearing in Part II. Their estimation was 78 FTEs for computer science.
- The application science participants were asked to estimate the minimum level of additional effort on data management that their program will be driven to provide to achieve its mission. That is "How many FTEs would your program really have to make available?" Their estimation was 30 FTEs from the science programs.

Given their origins, these two estimates must be regarded as complementary. Historically, even with these two sources of effort, there have always been major gaps in hardening and packaging as well as maintaining and supporting computer science "products." As the tables in Part II show, the computer science work is predominantly in the research and development stages. The science programs know from experience that their effort must go mainly into deployment and maintenance, with some hardening and packaging.

While vital, people are not always enough. The development of scalable approaches to high-volume data management is impossible without the availability of test facilities involving substantial hardware investments. Experience in data-challenged fields indicates that these facilities add about 50% to the development cost. Considering that information-intensive efforts have more modest hardware needs leads to an average increment of about 30%. Thus, the required scale of additional effort is about 108 skilled FTEs, plus test facilities, translating into a program of about $32M per year.

I-3.4 Developing the Office of Science Data-Management Program

An effective Data-Management Program requires the following actions:

- Ongoing assessment and ranking of efforts based on the evolving needs of the science programs
- Full exploitation of the considerable commonality between sciences to drive the development of tools that have wide applicability
- Setting of an appropriate balance between research and development, hardening and packaging, and maintenance and support
- Careful sizing of the program to optimize the long-term scientific productivity of the Office of Science

The SciDAC program was repeatedly identified during the workshops as having two related key aspects that *must* appear in a Data-Management Program for 21st-century science:

1. Cross-disciplinary collaboration as the foundation of most major projects
2. Joint application-science and computer-science funding of some major projects

SciDAC has already shown hundreds of scientists that cross-disciplinary collaboration is difficult but exciting and ultimately highly productive.

In large measure, the requirement for priority ranking based on the needs of the science programs can be addressed by ensuring that a large fraction of the funding, be it new or existing, flows through these programs to the cross-disciplinary projects they identify as important. This approach also ensures that appropriate attention will be given to hardening and packaging, maintenance and support, in addition to the computer-science research issues.

The ideal core approach thus becomes the following:

1. Provide additional data-management funding for both ASCR and the science programs, such that both can fully carry out their roles in the data-management program.
2. Require the majority of successful proposals to involve both funding and collaborators from ASCR and the science programs.
3. Provide oversight at the Office of Science level to ensure that the data-management funding is set at a level that optimizes Office of Science success and that solicitations result in projects that are appropriately forward-looking and interdisciplinary.
4. Appoint a Program Director with responsibility for the coherence of the program.

Part II: Data-Management Technologies and Gap Analysis

Part II of this report sets out the computer-science perspective on the exciting needs for data-management research and development that are driven by the requirements of the application sciences. The computer scientists were asked to characterize the maturity of each needed activity that they identified: was it at the pure research and development stage, or was it beginning to be focused on the later stages of a product lifecycle, such as hardening and packaging or even support and maintenance.

We were not surprised to find that the majority of activities were considered to have an initial focus on research and development. But prominent computer scientists stressed that working with application scientists to harden and generalize data-management tools was itself a productive area of computer science.

Each section in this part of the report concludes with a table listing the topics where work is needed and indicating whether the main focus is on research and development, packaging and hardening, or support and maintenance. The intention is that this material inform, rather than determine, the future process that will allocate Office of Science resources to work on data management based on evolving needs and opportunities.

II-1 Workflow, Data Flow, Data Transformation

We focus here on four areas of workflow: specification, execution, monitoring, and development.

II-1.1 Workflow Specification

Workflow management systems help in the construction and automation of scientific problem-solving processes that include executable sequences of components and data flows. In addition, such systems typically offer the following services:

- Automatic sequencing of component (or “operator”) invocation
- Component and flow synchronization
- Direction, control, and fail-over management of data flows between components (for example, through “background” data movers)
- Tracking and reporting mechanisms (process progress, auditability, provenance, quality)

The resulting gains in scientific productivity are comparable with the huge gains previously achieved by the introduction of database technologies that made components data-independent.

II-1.1.1 Current Status

In general, the workflow market can be divided into business-oriented workflow products and scientific workflow systems. Business-oriented products such as FileNet, Oracle Workflow, and IBM’s MQ Workflow are used mainly for document distribution, business processes, and e-commerce. In comparison, scientific workflow systems operate on large, complex, and heterogeneous data; can be computationally intensive; and produce complex derived data products. Scientific workflow systems often need to provide for load balancing, parallelism, and complex data flow patterns between servers on distributed networks. To date, very few scientific workflow products have been produced, and these are mostly academic and in their experimental stage and not used on a large scale. Examples of such systems include Ptolemy/Kepler [Kepler], SCIRun, Triana, Taverna, and commercial systems such as Scitegic/Pipeline-Pilot and Inforsense.

Workflow specifications can be described by using a number of different layers (see Figure II-1.1). The layers represent different aspects of the workflow, such as control flows, data flows (including I/O), event flows, software components, computational elements, and storage components. For example, the control layer allows the workflow to describe the sequence of tasks to be performed (expressed as actigrams or datagrams), where each task can invoke one or more software components.

> **1. Flow layer –** This layer describes execution ordering of tasks by using different views of sequencing, branching (decision making, parallelism), and feedback (loops)—different constructors that permit flow of execution control. Examples of constructors are sequence, choice, parallelism, and join-synchronization. Tasks in their elementary form are atomic units of work; they may also invoke other

applications and/or tools. In compound form, a task can be a subworkflow, that is, a module consisting of an ordered execution of a set of tasks.

2. Application and Software Tools layer – This layer describes the invoked applications and software tools used by the workflow tasks. In most cases there is a one-to-one correspondence between tasks and invoked applications. Additional narrative explanations can also describe the invocation mechanism (i.e., CORBA,[7] Web services).

3. I/O System layer– This layer describes the I/O systems that allow efficient read and write operations by the applications. Predicted data volumes and their characteristics, such as streaming granularity, can also be described on this layer.

4. Storage and Network Resource layer – This layer provides information about physical devices used by the tasks during their executions. The information includes performance-related issues such as the required data transfer rates.

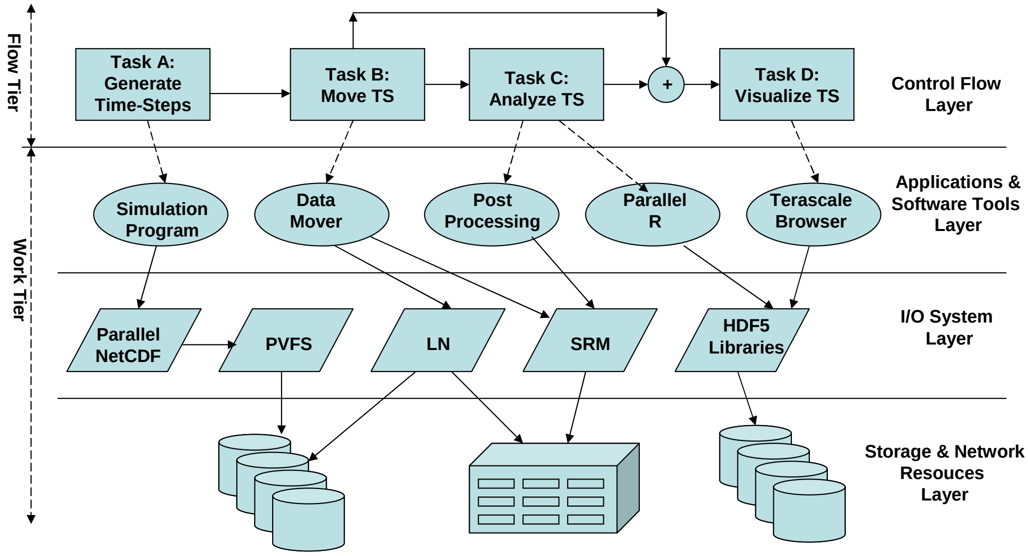


Figure II-1.1 Anatomy of a scientific workflow-management system

II-1.1.2 Gaps and Needed Research

Graphical representations of control and data flow in scientific workflows have scalability limitations as the number of components becomes large. A scientific workflow language is needed that can describe the following effectively:

- Inputs and outputs for each components
- Metadata of the workflow
- Granularity of tasks, subworkflows

[7] Common Object Request Broker Architecture; see the Object Management Group's website: http://www.omg.org/

- Task invocation – Web services, CORBA, wrappers, callbacks
- Human tasks: notifications and alerts, steering
- Data-flow streaming granularity
- Performance expectations

An emerging area of study, similar to the extensive work done in software patterns, is workflow patterns. This area grew out of the Ph.D. work of Kiepuszewski in 2002. It is reminiscent of the EDSS DAG patterns but is more elaborate. Today, it is primarily the work of Wil van der Aalst and collaborators [Aalst00], who have expanded concepts in graph theory to understanding workflows and workflow languages. The contributors to workflow patterns lament that workflow products have been created without much thought to their theoretical completeness or formal capacities. The workflow patterns group, therefore, seeks to refine the construction of workflow languages into a scientific process. The group does not advocate any specific language but is exploring Web Service Composition Languages as a subset of workflow languages.

II-1.2 Workflow Execution in Distributed Systems

In order to prepare for the future needs, the framework implementation must be scalable and be able to send data from thousands of heterogeneous processors to other receivers, which can be from one processor to thousands of processors. Otherwise trivial overheads can become significant on the scale of thousands of processors. Where the data source is not reproducible (e.g., time-dependent astronomical sky surveys) or simply very costly, the implementation must be able to accept the full flow of data. Optimizing the load balancing of all elements is a particular challenge in a distributed system.

The handling of security in the workflow management system involves two separate issues. First, the workflow management system should provide access controls to the scientists and their collaborators that limit access to the scientists' specific workflow. Only the collaboration group should be able to create, modify, and monitor their workflow. Second, the workflow will need to hold the credentials of one or more of the collaborators to enable the various workflow components to access the necessary resources. The workflow management system will need to protect the credentials while they are in the custody of the system.

Data marshaling in particular is something of an orphan: the components themselves should not need to know that they might be using data from or sending data to remote distributed or parallel components. If they did, there would be an explosion of complexity inside the components. Thus, data marshaling must take place in the framework. Existing component frameworks such as CCA, however, do not address data marshaling.

II-1.3 Monitoring of Long-Running Workflows

The benefits of workflow monitoring are numerous:

- Predicting the future behavior of a running workflow
- Enabling flexible decisions and deadline-miss prediction
- Providing knowledge about the current workflow (state, data, and timing)
- Supporting active notification about certain conditions

- Enabling dynamic workflow optimization
- Allowing for off-line workflow analysis applicable to local and distributed workflows

II-1.3.1 State of the Art

Workflows are monitored in both online and offline mode. Online monitoring is performed while the process is running and is designed to provide information about the current state of local and outsourced workflows (for user, application, and other modules). Online monitoring typically generates log files for analysis and offline monitoring. Offline monitoring is performed by analysis of the log files and can help in workflow optimization and failure detection and recovery.

II-1.3.2 R&D and/or Deployment Needed

Complex scientific workflows will often run for long periods and thus will need to be able to recover from dynamic changes. The first issue is how to handle well-defined fault conditions such as a server that is down or a resource that cannot service the request at that moment (e.g., it is out of storage space) when the workflow manager tries use the resource.

The second issue is how to treat failures that render resources (temporarily) inaccessible, but not necessarily inoperative. The problem of monitoring resources accessed over wide-area networks is an example. If the network between the monitoring service of the workflow management system and the resource that is acting on behalf of the workflow partitions (no communication is possible between the two halves), then the monitoring service cannot determine whether the resource is still acting on behalf of the workflow manager.

Users must be able to specify and optimize the policy to be applied in both of these situations: to decide whether the workflow should be restarted using different resources or to wait for the original resource to become available again or both.

II-1.4 Adapting Components to the Framework

We must allow for easy integration of existing data-mining algorithms facilitating feature extraction and tracking. These data-mining modules should take advantage of the common data model of the framework. In addition, we must allow for the easy integration of legacy codes as well as for new high-performance versions of legacy codes that ultimately are created. Such integration will require the development of a workflow framework with interface standards to enable workflow composition, automation, interoperability of workflow components, and extendibility. Integration often involves the creation of “adaptors” and “bridges” that allow for interoperation between technologies of different provenance.

II-1.5 Summary Table

Table II-1.1: Summary of the major issues that require further research and/or deployment efforts in the area of scientific workflows. The ellipse marks the current position of the issue relative to pure R&D all the way on the left and off-the-shelf deployment all the way on the right.

Issues	**Research and Development**	**Hardening and Packaging**	**Deployment and Maintenance**	**Comments**
Granularity of tasks, subworkflows		⬭		
Task invocation		⬭		Wrappers for legacy codes
Human tasks: notifications, alerts, steering	⬭			
Data-flow streaming granularity	⬭			Research needed both in the specification and in the implementation phases
Performance expectation	⬭			
Workflow engine for scientific applications		⬭		
Integrated data-flow management	⬭			
Failure detection and recovery	⬭			Especially needed for distributed and long-running workflows
Data-driven flow control		⬭		
Performance-driven flow control	⬭			
Workflow optimization	⬭			
Run-time resource coordination	⬭			
Workflow Security		⬭		
Data marshaling	⬭			

II-2 Metadata, Data Description, and Logical Organization

This section focuses on keeping track of data. High-throughput techniques and increasingly complex analysis workflows produce tremendous numbers of raw and derived datasets, causing an explosion in the number of datasets that must be managed within a single project. Management of data at the community level, and even across communities, is also a growing issue. Understanding complex real-world systems can require access to information from physics, chemistry, biology, environmental science, and more.

We must enhance our ability to describe data and their relationships with other data—the logical organization—and to use that description to discover, interpret, evaluate, and transform the data. This additional description is often referred to simply as "metadata," and managing such information is considered "semantic engineering," or "knowledge engineering."

Several promising lines of research and development and a number of pilot and project-level deployment efforts are available to guide the development of robust semantic data infrastructure. Conceptual models such as controlled vocabularies, schema, and ontologies provide increasing levels of description based on agreements concerning the meaning of terms, allowable data hierarchies, and the overall data model, respectively. Associated standards ranging from self-describing data formats and schema languages to ontology and inference languages allow these concepts to be recorded in human and machine-readable forms. Basic tools have also emerged for creating common descriptions, capturing metadata for individual datasets, and storing, viewing, querying, and making inferences from metadata.

Nonetheless, significant gaps remain between current capabilities and the semantic cyberinfrastructure needed for next-generation scientific data management. Many of the gaps relate to the scaling of technologies. Phenomena studied in science span tens of orders of magnitude in size and duration. Scientific semantics, the definition of models and theories, are extremely precise. Some of the most challenging research projects involve semantics from multiple domains as simple as using nanoscale photon detectors to image galaxies or as comprehensive as investigating genetic determinants of disease and evaluating potential cures. Existing semantic technologies have never been deployed into an environment with so many simultaneous challenging requirements.

For scientific use, the metadata infrastructure will need to be general and extensible enough to support arbitrary domain-specific metadata definitions and to support the evolution of these definitions over time. The infrastructure must support users across domains and virtual organizations in sharing data and metadata across multiple ontologies. Given the scale and robustness requirements, the infrastructure must be able to support distributed and replicated metadata stores that may be optimized for specific purposes, while still supporting federation across heterogeneous stores to enable complex data discovery operations, long-term digital preservation, and other global processes.

II-2.1 Data Models and Formats

Data models and formats are defined early in the scientific lifecycle. Models are chosen to represent the phenomena under study and facilitate the testing of hypotheses. Formats are then chosen based on a combination of factors involving decisions about ease of use, efficiency, and compatibility with the existing infrastructure. Descriptions of the formats and models, which may be as limited as the use of well-known extensions on file names, convey these decisions to colleagues and software through other stages of the lifecycle.

More formal descriptions reduce the expertise needed to understand data content and enable the development of more general software tools and increased automation of data flows. Common data formats, such as GIF and JPEG image formats, help encode best practices and allow some level of software reuse. Self-describing formats, such as the FITS format used by the astronomy community, the netCDF format used by the climate community, HDF, and more recently XML, enable richer sharing. For example, by defining common vocabulary or schema (e.g., using XML Schema), groups can define new common models without requiring new software for reading and writing files. Further, by their nature, self-describing formats support data inspection with generic tools. For example, using the Chemical Markup Language (CML), researchers can use standard tools to display chemical information such as three-dimensional molecular structures and can use standard schema validators to verify that a given dataset contains required information and is correctly structured. Beyond schema, modeling languages and ontologies formally describe relationships among data elements, allowing additional automation. For example, ontology languages such as OWL[8] can be used to state that two units (e.g., “foot” and “meter”) are both units of linear measure, reusing existing, proven ontologies in ontology repositories. The statement can then be used to infer an automated translation between data using these units given a conversion factor. The use of ontologies in science is still in its infancy, but two examples are the Microarray and Gene Expression (MAGE) standards in biology.

All of these technologies are primarily prescriptive: the decision to use them must be made before data can be recorded, and data acquisition and analysis tools must be built to support them. An alternative approach is to provide a machine-readable description of the format or model that can be applied to existing data to allow it to be interpreted in terms of common vocabulary, schema, or ontology. Scientific examples include several related approaches currently being standardized as the Data Format Description Language (DFDL) within the Global Grid, which targets description at the level of schema.

While a strong base exists in this area, numerous challenges remain for meeting the needs of next-generation science. Although various tools exist for defining schema and ontologies (e.g., xmlspy,[9] Protégé,[10] and the Ontolingua[11] development tool), they are

[8] OWL: http://www.w3.org/TR/owl-features/

[9] Xmlspy: http://www.altova.com/products_ide.html

[10] Protégé: http://protege.stanford.edu/

[11] Ontolingua: http://www.ksl.stanford.edu/software/ontolingua/

aimed primarily at knowledge professionals. Easier-to-use tools are needed, for example, tools that guide researchers in developing or selecting ontologies and relevant pieces of ontologies as part of an overall research protocol development activity. Further, given that use of schema and ontologies both formalizes and increases the amount of information being recorded, capabilities for automating the capture and validation of information become critical. To enable such automation will require that the scientific cyberinfrastructure become metadata aware, allowing tools that generate metadata (domain software as well as middleware involved in data flow) to communicate it to tools that store, manage, and use it.

Aside from these usability issues, there are also concerns that current tools will not be capable of handling formats and models as complex as those used in science, ranging from the ability to intelligibly display large models to maintaining performance when faced with large models and large data. Evolving formats and models, which is central to scientific progress, is also not addressed well in current systems.

II-2.2 Managing Metadata

During later stages of the scientific lifecycle, researchers need to recall appropriate subsets of their data for analysis and then may need to translate data into new formats and models, fuse data from multiple techniques, and publish derived results in other formats. As noted throughout this report, challenges arise in automating these steps in an efficient and cost-effective manner. Common formats and models and self-described or externally described formats are a critical foundation for higher-level query, translation, and workflow mechanisms and integrated user environments. All of these techniques, to varying degrees, decouple the domain-specific aspects of data (i.e., the meaning of the data) from its logical description and organization. Thus, issues related to efficiently working with large volumes of data can be tackled across domains and packaged in common programming interfaces and protocols.

Currently, metadata is managed in a variety of ways: filenames and directory hierarchies, tagged information within files (e.g., XML, OWL), relational databases, metadata catalogs, and triple stores,[12] as well as combinations of these. Metadata may be managed by multiple systems specialized for specific uses (e.g., replica catalogs, annotation servers) or combined in one (e.g., by using the WebDAV[13] protocol, which supports storage and retrieval of arbitrary metadata). All of these approaches have advantages, and future metadata management systems probably will need to span them. Further, future metadata services must provide federation across distributed, independently managed metadata systems and among heterogeneous metadata ontologies (as well as versions of individual schema and ontologies).

A variety of metadata catalog technologies are available, including the Globus Metadata Catalog Service [Singh2003], the Storage Resource Broker's MCAT metadata catalog

[12] See, for example, http://www.w3.org/TR/rdf-concepts/

[13] WebDAV: Web Distributed Authoring and Versioning, http://www.webdav.org/

[SRB], UCAR's Thematic Realtime Environmental Distributed Data Services (THREDDS) [Domenico2002], and custom-designed, application-specific catalogs used by communities such as the Earth System Grid, the European Data Grid, and the Laser Interferometer Gravitational Wave Observatory project. Some, such as the Scientific Annotation Middleware[14] (SAM), are exploring services for automated metadata extraction from files and metadata translation. Standards for interoperable metadata services do not exist outside specific areas (e.g., The Distributed Annotation Service, or DAS,[15] used in biology and the Replica Location Services standardized by the Global Grid Forum).

Richer metadata services must be developed to support a mixture of file-, database-, and catalog-based systems. These services will need to provide robust, high-performance fault-tolerant capabilities through techniques such as clustering, replication, and synchronization. While these features sometimes exist in file and database-level systems, most current catalogs are based on a centralized metadata repository for ease of maintaining consistency. Also needed are enhanced capabilities for supporting discovery and queries spanning schema and ontologies, managing their evolution, and controlling access to individual types of metadata.

II-2.3 Using Data Descriptions and Relationships

The term *metadata* includes not just the description of the contents of a dataset but also information about the relationships between datasets. Often, the logical organization of data—its relationships—is defined in terms of the processes in which these relationships are generated or used, for example, workflow/provenance, project/records management, annotation, and discovery.

Scientific workflows consist of experimental data collection, simulation, and/or analysis tasks. Provenance information is metadata that describes the logical organization of data in terms of its origins, including the original conditions under which an ancestor dataset was produced, the sequence of transformations applied to produce the derived data, and the people and software involved in performing these transformations. Provenance includes description at the level of science (dataset A is the Fourier transform of dataset B) and engineering (the transform was done with version 2.3 of software package X on a specific compute resource). Provenance metadata, particularly engineering-level information, is most easily collected directly from applications and workflow systems and can be used to create new, related workflows, for example by using the provenance of one analysis pipeline to instantiate a parameterized analysis of additional datasets.

An example of a workflow management system that captures provenance information is the Pegasus[16] system for planning and execution in Grids, which was developed as part of

[14] Scientific Annotation Middleware: http://collaboratory.emsl.pnl.gov/docs/collab/sam/

[15] Distributed Annotation Service: http://biodas.org/

[16] Pegasus: http://pegasus.isi.edu/

the GriPhyN[17] (Grid Physics Network) project. The Pegasus system takes a high-level definition of a desired workflow and schedules the tasks in the workflow on available resources, based on the requirements of the tasks and the availability of resources in the Grid. Pegasus tracks the original abstract workflow, the input files, and the output files that are generated as products of the workflow execution. Pegasus highlights the fact that datasets are not the only entities that will require metadata descriptions; it will be necessary to describe hardware and software tools with information about their provenance and the data they are capable of processing. Another example is the MyGrid[18] system, which enables a biologist to dynamically compose workflows and discover quickly sequences of interest among the thousands returned by curated databases for an investigation.

Tools such as problem solving environments, portals, and electronic notebooks can also document aspects of workflow, but they are more directly involved in the logical organization of data into project and experiment hierarchies. These tools can also be used to support a wide range of structured and unstructured annotations, such as a similarity between a gene in one organism and one in another, information about a detected feature, reviews of data and assertions about data quality, or simply some text about an idea for a new experiment triggered by current work. For example, the SAM-based Electronic Laboratory Notebook allows text, drawings, images, equations, and arbitrary files to be associated with data and organized into electronic chapters and pages.

As with workflow, these other kinds of metadata can be used within corresponding processes, for example, reporting project progress, assembling legally defensible records of work, and aiding in data discovery. One also can combine types of metadata to support advanced queries. For example, scientists might search for potentially misidentified features derived from data from a specific instrument during a time period when, as is later discovered in an instrument log, it may have been miscalibrated.

Despite successes in specific communities, use of metadata to support scientific processes is far from ubiquitous. As noted in previous subsections, numerous issues related to the capture and management of metadata need to be addressed to enable automation and integration of the types of functionality described here. In addition, more work will be needed to define the types and granularity of metadata, such as provenance, that should be captured, that is, that will provide sufficient value to justify the cost of their capture and management. Given that the analysis of cost/benefit ratios may show domain-specific results, ontology research will also be required to provide the correct level of detail for the desired capabilities and to categorize these capabilities into a set of general metadata services maintained as cyberinfrastructure.

[17] GriPhyN: http://www.griphyn.org/

[18] MyGrid: http://www.mygrid.org.uk/

II-2.4 Summary Table

Table II-2.1 Summary of the major issues that require further research and/or deployment efforts in the area of metadata, data description, and logical organization. The ellipse marks the current position of the issue relative to pure R&D all the way on the left and off-the-shelf deployment all the way on the right.

Issues	Research and Development	Hardening and Packaging	Deployment and Maintenance	Comments
Automated capture and validation of metadata	⬭			Addresses metadata for legacy and future data.
Self-describing data formats and models		⬭		Supports the development of standardized data models in each domain
Data description development tools/services	⬭			Addresses ease of use, graphical representations, granularity and scalability of schemas, data models and ontologies
Inference engines, and metadata translation tools/services	⬭			Includes mappings, schema translations, and schema evolution.
Languages for schemas, data models, and ontologies		⬭		Standards developed by the W3C and GGF must be adapted for scientific community
Semantic models for workflows and integrated environments	⬭			Standardization of provenance, analysis and visual integration models (see Sections II-1 and II-2 in particular)
Provenance-tracking tools/services	⬭			Reusable tools across applications
Scalable, distributed repositories for data models and associated tools/services	⬭			

II-3 Efficient Access and Queries, Data Integration

Most of the world turns first to commercial database management systems whenever there is a need to access or query nontrivial amounts of information. Scientists are quite happy to use database technology to handle simple tables and other database applications. Rarely, however, do they devote much effort to addressing their challenging data-management needs with database technology alone.

Why is this so? First, the database industry has often turned its back on scientific problems at a scale of 10 to 1,000 times those encountered by leading commercial customers. Second, database management systems normally provide features such as transactions and highly granular locking that are largely unnecessary for read-dominated scientific applications.

One example of simultaneous success and failure is the use of Objectivity DB to store and access almost a petabyte of data from the BaBar high-energy physics experiment at SLAC. This deployment, particularly the five-year joint SLAC-Objectivity Inc. work on scaling issues, was a major technical success. Indeed, Objectivity Inc. attribute much of their current business to the capabilities developed and hardened during the work with SLAC. Nevertheless, BaBar has now largely abandoned Objectivity DB in favor of the high-energy-physics community code ROOT, which embodies a minimalist set of object persistency and access mechanisms and is perceived by scientists as much simpler.

If, in addition to advancing database technology for commerce and national security, the BaBar-Objectivity experience had also been perceived as bringing even a small net benefit to individual scientists, the experience would have strongly supported a major thrust focused on the use of database management systems at the leading edge of scientific data management. Even with the experience of partial failure, there is a strong case for promoting contact among data-intensive science, computer scientists focusing on database issues, and the commercial database industry. This contact should extend to trial deployments, in judiciously chosen areas of data-intensive science and database technology.

The next two subsections address two of the key technologies needed for scientific data access and querying. The third subsection then describes issues of data integration.

II-3.1 Large-Scale Feature-Based Indexing

An effective indexing scheme can speed many data analysis tasks. To illustrate the challenges, current approaches, and potential solutions, we give two examples: searching large high-dimensional datasets and identifying regions of interest.

Many large datasets contain a large number of attributes. For example, a typical high-energy physics experiment produces a summary dataset with 500 searchable attributes for billions of collision events. To find interesting events, physicists may apply range conditions on a handful of attributes, such as “numberOfTracks > 1000 and numberOfAntiParticles > 3.” Efficiently answering these partial-range queries is a serious challenge. The traditional indexing techniques, such as B-trees and hashing, are inefficient for datasets with a large number of searchable attributes. Even multidimensional indexing techniques, such as R-trees, are efficient only for datasets with no more than 10 or 15 attributes. If there are more attributes or if the user query involves only a small number of the indexed attributes, a brute-force scan is more efficient than these indexing schemes.

A typical scientific computation, such as computing the heat generated from an ignition kernel, requires the identification of features known as "regions of interest." Usually a region of interest is identified in two steps: a searching step to find all the objects satisfying some user-defined conditions and a region-growing step to group the objects into connected regions. Most approaches partition data according to spatial attributes. Since the conditions usually also involve other attributes, these approaches are essentially performing a brute-force scan in the searching step. A number of researchers have proposed using database indexes to speed the searching step; however, such schemes slow the region-growing step because they break apart neighbors on the underlying mesh.

Another possible technology is the bitmap index, which is effective for scientific datasets that typically are read-only or read-mostly. Bitmap indexing has been shown to significantly outperform all other indexing techniques for partial-range queries. In particular, bitmap indexes are efficient in identifying features on regular meshes. Since the bitmap index does not reorder the data, it can speed the searching step without slowing the region-growing step. One high-energy physics experiment is working on its own version of bitmap indexes, and another one is planning to do so.

Effectively searching over billions of small objects is a problem facing many scientific applications.. To be more useful, however, the bitmap index approach needs to be extended to work with more complex meshes, such as those from adaptive mesh refinement. Since both searching and feature identification are typically performed as parts of a larger analysis process, an important research and development issue is to seamlessly integrate the bitmap indexing software with other analysis and visualization tools.

II-3.2 Query Processing over Files

Traditional databases (e.g., those based on relational or object models) provide capabilities such as the ACID[19] properties, which make them attractive for transactional or ad hoc analysis queries. Such database systems hide from the user the details of most underlying operations, including I/O, storage, access strategies, indexing, and data movement. These databases tend to be "heavyweight," and their designs are generally optimized for workloads quite different from those found in scientific applications.

In scientific data-management systems, many applications require capabilities to perform queries and analysis over a very large number of (large) files. The structure and content of these files depend on the application domains they come from. For example, files

[19] ACID: **A**tomicity of transactions, **C**onsistency of the database after every transaction, **I**solation between simultaneously requested transactions, **D**urability of information committed to the database by successful transactions.

containing climate modeling data may be in NetCDF format, whereas data from a high-energy-physics application may be in a specialized format called ROOT. Thus, the functions and programs to perform analysis and queries are written by using the interfaces available with those formats. Furthermore, parallelization and scalability for these functions, including I/O operations, are critical for obtaining good performance.

Figure II-3.1 illustrates the functionality envisioned by a scientific database system. The system must have capabilities for managing metadata generated and derived from scientific data, a query and analysis capability on these metadata, and the ability to manage a large number of files with associated query functions.

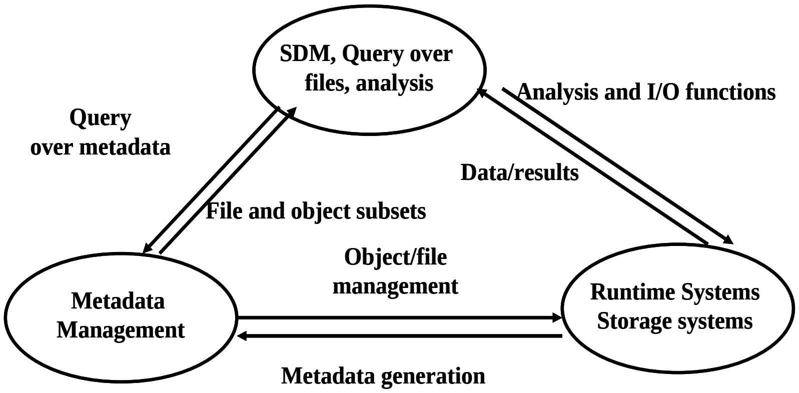


Figure II-3.1 Illustration of functionality required for query processing over files

The metadata management component entails use of a "lightweight" database system (e.g., MDMS [Chou00, No03]). Such a system is responsible for storing and managing metadata and the results of queries. The runtime systems component refers to libraries for data access and manipulation supporting high-performance I/O as well as various interfaces related to the particular application domain. Thus, users can continue to use access functions from their application domain without being constrained by the database system. Queries and analysis functions, part of the third component, permit users to develop applications by using standard interfaces and libraries.

Scientists need a unified query interface that will support queries over data and metadata in different formats, with implementations that are reusable in many contexts. In addition, these tools can also benefit from the provision of reusable facilities for buffering, caching, and indexing scientific information. Together, these facilities will move us closer to filling the following large gaps between the current available scientific data query facilities and the growing user requirements:

- *Dealing with heterogeneity and legacy data.* Advanced data integration techniques are required to successfully use the increasingly rich collection of shared scientific data.

- *Interdisciplinary data sharing*. The same datasets may be used very differently by different user groups. For example, the neutron scattering data produced by SNS will be postprocessed in diverse manners by scientists from the fields of material science, physics, chemistry, and biology.
- *Integrated data query and storage services.* In a transparent, query-enabled distributed data environment, storage and data access services should be integrated with data query facilities, in order to efficiently replicate and retrieve *interesting* data for users.

Some tools do exist for interactively browsing datasets. For example, the Scientific Data Browser developed at NCSA [SDB] enables users to use a Web interface for browsing through annotated scientific datasets written in popular formats such as HDF and NetCDF. Another example is the GODIVA framework developed at UIUC [Ma04], which provides lightweight database support to help visualization tool developers organize and search for in-memory datasets.

How to provide data-format-independent, application-tailorable facilities for buffering, caching, indexing, and querying information is an open research problem in the database research community. The first step in answering this question is to investigate what the interface (API) should be for access to these facilities. The second step is to federate and integrate such facilities across multiple data and metadata sources. The third step is to integrate the new data query facilities with large-scale data storage services, in order to provide seamless location of interesting data.

II-3.3 Data Integration

In answering fundamental questions about natural phenomena, application scientists routinely deal with multiple data and information sources. For example, to identify and characterize regions of functional interest in genomic sequence requires full, flexible query access to an integrated, up-to-date view of all related information. However, the dramatic growth of scientific data sources has made the task of finding, extracting, and aggregating relevant information extremely difficult because of a number of factors. First, the data sources are physically distributed and heterogeneous in how information is stored, organized and managed. Second, they reside on heterogeneous hardware platforms with diverse software interfaces. Third, the data is of different types (e.g., text, video, images, audio) and formats (e.g., netCDF, HDF, SILO) as well as dynamically changing in both content and form.

Data integration aims to provide users with a uniform interface to access, relate, and combine data stored in multiple, geographically distributed, and possibly heterogeneous information sources. It enables users to focus on specifying what they want, rather than thinking about how to obtain the answers. Consequently, users are freed from the tedious tasks of finding the relevant information sources, interacting with each source in isolation, using a particular interface, and combining data from multiple sources.

II-3.3.1 State of the Art

Data integration requires resolving the differences and inconsistencies in the data management systems (e.g., different vendors), in the data models (e.g., relational, network, ER, object-oriented), in the query and data manipulation languages, in the data types (e.g., text, graphics, multimedia, hypermedia), in the format (e.g., structured, semi-structured, specialized formats), and in the semantics. The ability to manipulate data requires both a characterization of the internal structural and semantic properties and a characterization of the relationship of the dataset to the associated material. Moreover, to achieve the vision of data integration, one must have the technology to describe the data models, data structure, data format, and data semantics.

Given that the support for such metadata descriptions exists, we can take two general approaches to providing data integration: data warehousing and federated databases.

In *data warehousing*, information from each source from the domains of interest to specific users is extracted in advance, translated and filtered as appropriate, merged with relevant information from other sources, and stored in a (logically) centralized repository. When a user poses a query, the query is evaluated directly at the repository without accessing the original information sources. Data warehousing has been extensively researched in the context of analytical processing for decision support in the business domain, and the technique has, in general, been successful for domains where the system is specialized to the application. There remains, however, a need for general-purpose scientific data warehousing tools that can support structured data as well as files in a uniform way based on the metadata descriptions.

In the *federated databases* approach, the user query is decomposed and sent by the data integration system to appropriate information sources that can answer the query. Once the partial results are obtained, the system performs the appropriate translation, filters and merges the information, and returns the final answer to the user. Federated databases have had some success, especially when the systems being federated are similar and support the same data model (such as federating relational database systems). However, their use with heterogeneous data systems, data models, and data formats that exist in scientific domains is still an open research area.

II-3.3.2 Gaps and Research Needed

In order to advance science, infrastructures are needed that enable not only using but also fusing information from multiple scientific data sources from multiple digital archives. The integration should be built both within and across boundaries of established disciplines. Such integrated infrastructures should embrace existing but complementary data organization mechanisms supported by digital libraries, data Grids, and persistent archives. They should be scalable in terms of the size and number of databases and should be supported by efficient underlying storage and file systems. They should be based on standards, to the extent possible, and should be driven by domain-specific ontologies. Data integration technology should be readily used by workflow systems. The combined system should support data ingestion from remote sensors and experimental devices, publication into collections, analysis on compute platforms (possibly specialized), comparison with simulations, and archiving for long-term preservation. Hence, the data integration must become an integral part of workflow processing environment.

II-3.4 Summary Table

Table II-3.1: Summary of the major issues that require further research and/or deployment efforts in the area of efficient access and queries, data integration. The ellipse marks the current position of the issue relative to pure R&D all the way on the left and off-the-shelf deployment all the way on the right.

Issues	**Research and Development**	**Hardening and Packaging**	**Deployment and Maintenance**	**Comments**
Large-scale feature-based indexing			⬭	Feature-based refers to indexes for searching over multiple features concurrently
Integration of indexing technology with analysis and vis		⬭		
Lightweight data querying from large datasets		⬭		Lightweight refers to partial data management capabilities, such as excluding locking and recovery
Querying files in multiple data formats	⬭			
Integrated data query storage services	⬭			
Integrated metadata management and file management	⬭			Metadata refers to information about the content in files
Specialized scientific warehousing		⬭		
General-purpose scientific data federation	⬭			
Data integration in workflow systems	⬭			

II-4 Distributed Data Management, Data Movement, Networks

The management of distributed data raises numerous challenges. The transfer of such data requires attention to data placement and replication, data flow, and multiresolution data movement. Effective management of distributed data also places demands on the network infrastructure. Moreover, the issue of security becomes paramount whenever large amounts of data must be transferred. In this section we address each of these concerns.

II-4.1 Data Placement

In the same way that the load register instruction is the most basic operation provided by a CPU, so is the placement of data on a storage device the foundation on which any data management system resides. Data placement consists of two elements: selection of an available storage location capable of holding the data (we refer to such a location as a "lot" [Bent2002]) and the actual transfer of the data into this location. Data placement is thus a two-step operation. First, an appropriate lot has to be secured; only then can the data be copied from its current location to the allocated space. Since both steps require the allocation of resources, they must be treated as individual tasks each of which may experience arbitrary delays in execution. Moreover, they can also fail or be denied execution altogether. It is therefore essential that at all levels data placement tasks be treated in the same way computing tasks are treated [Kosar2004].

Regardless of whether it is an end user, an application, a middleware component, or a low-level system function, the entity that triggers a data placement request must be able to influence when, where, and for how long the data should be stored. The decision can be based either on the properties of the target storage unit (e.g., proximity to the current data location, reliability, throughput) or on a set of goals or intentions (e.g., keeping the checkpoint data until the end of the simulation run). Providing data placement services requires an appropriately layered design.

Currently, a plethora of mechanisms exist for moving data from one storage device to another. Only few systems provide mechanisms for data placement, however. File systems treat data placement as a side effect of creating a file or writing to a file. In fact, some file systems support a user-based quota mechanism that prevents a user from storing more than a predefined amount of data. Within a single system, file systems are the default provider of data placement mechanisms and generally offer no user control of placement. Within an organization, mechanisms vary widely from network file systems (e.g., NFS and AFS) to hierarchical media (i.e., disk and tape) file systems (e.g., HPSS). In wide area networks, data transfer mechanisms include file transfer tools (e.g., GridFTP) and data block servers (e.g., IBP or iSCSI). The different mechanisms provide a spectrum of utilities capable of controlling properties like access times, throughput, reliability, and scalability. These mechanisms should be available to scientists when they need fine-grained control, albeit with a tradeoff in complexity of use.

On the other hand, scientists who does not need such fine-grained control should be able to invoke a policy layer that will implement their intentions. For example, within a single simulation run, a scientist may wish to store checkpoints within the supercomputer for the duration of the next time-step, save all the output of every time-step, and select regions of data in a remote storage system. The need for checkpoints requires storage that has low latency, high-throughput, and short duration. The need for the entire output data requires storage with high reliability for long duration (in this case, latency and throughout are secondary). The need for select regions of data stored remotely requires low latency and high throughput to the remote user (who may be the same application scientist). Although many mechanisms exist for data transfer, research and development is still required to create schedulers and planners for storage space allocation and the transfer of data.

II-4.2 Replica Management and Movement

Replica management and cache management are closely related, but replica management focuses on the particular issues that arise in the management of geographically distributed copies of datasets. In geographically distributed computing environments, computational tasks may be performed at locations that are far away from necessary datasets [CDF+02, KLSS03, Atkinson2003, Chervenak2004]. In such cases, remote data access can be orders of magnitude slower than access to a local file. Replication involves creating multiple copies of identical files or portions of files in order to increase data locality and fault tolerance and to reduce the latency of data access in a wide-area, distributed computing environment. Traditional replica management for transactional database management systems keeps track of table updates and synchronizes the changes among the database replicas. In scientific applications, most datasets are read-only after they are published, and data access is predominantly file-based; these characteristics simplify replica management because update synchronization is not needed.

Conceptually, replica management for read-only data is quite straightforward. It involves copying one or more files and registering them in a replica-tracking catalog or using a file representation that supports internal replication [SSS04]. In practice, however, specifying the precise replication actions needed to enhance performance in a given situation requires careful analysis because these actions depend on factors such as the overall schedule and priorities of outstanding requests, security and resource access policies, and the current state of distributed system resources. Specifically, the following issues need to be addressed: (1) specifying the source files to be copied and registered; (2) specifying the target directory or locations for the data; (3) specifying the catalogs in which new replicas should be registered; (4) coordinating copy and registration operations; (5) identifying and recovering from failures; (6) considering the state of resources, including network performance, existing replica locations, and the availability and performance of storage systems and computational resources; and (7) considering policy issues, including security and resource management policies that define which groups and applications have permission to access particular datasets, storage systems, and computational resources and what priorities are assigned to different requests.

An important task of a replica management system is *replica selection*: choosing among available replicas the one that will provide the best performance. In a distributed environment, replicas often reside on storage devices with access latencies ranging from minutes up to several hours. For instance, some replicas might be stored on a disk cache, while others are stored on tape drives. In addition, the network connections between the storage systems and computational nodes can vary significantly depending on the theoretical throughput of the network and the current network load. A replica manager should select the best replica with respect to specified criteria, such as minimizing access latency or maximizing overall distributed system throughput. This selection requires monitoring tools that track the network throughput and latency, storage system availability and performance, and the current load of CPU resources.

File replication arises in various scientific use cases. Typically, scientific datasets are stored in files and organized in a directory structure. In the most common use case, a scientist specifies a source physical directory and requests that the entire directory be copied to a target physical directory [SSS04]. A variation of this use case is to select and replicate files from the source directory according to some pattern, such as pattern matching on file names. In another variation, the scientist specifies explicitly one or more files or portions of files to be replicated.

Each use case raises requirements for significant research, both in the theoretical computer science of queuing and scheduling and at the systems and application integration level.

At the basic systems level, replica management services need to use the best available techniques for robust, high-performance data transfer: in particular, parallel streaming protocols with efficient and automatic recovery and restart of failed transfers. These systems must handle collections of files as a single request, and they need to operate in a manner that supports the security model of the wide area environment. Technologies such as the GridFTP protocol [GridFTP2004], the Reliable File Transfer Service [RFT2004], and Logistical Networking [PADB03] have made significant inroads in this area, but further work remains to make these tools faster, more powerful, and automatic.

Replica catalog mechanisms such as the Replica Location Service [CDF+02, Chervenak2004] have applied innovative techniques in cataloging structures that enhance speed and overall system reliability when tradeoffs can be made in probabilistic rather than transactional integrity of returned results. Such approaches have proven well suited to wide-area computing environments. The next steps of research in cataloging require improvements in the robustness, scalability, and self-organization of distributed replica catalogs [Cai2004]; management of flexible, distributed name spaces as a higher layer above the raw catalog mapping service; and integration of virtual-organization and group-cognizant authentication and authorization models into replica catalogs.

At the next level up, replication management needs to be integrated into a wide-area distributed workflow management system such as those being developed in Grid projects (e.g., PPDG, GriPhyN [Deelman2002, RF02], iVDGL, EEGE, and LCG). At this level, the work of replication needs to be both preplanned and request-triggered. The state of replication within a wide-area computing environment is a significant factor in the performance of a given workflow. Great opportunities for speedup exist by prestaging replicas to places where they will be most likely to offer workflow planners [Deelman2002] the opportunity to collocate computations with the datasets they require. Research on such wide-area prestaging, called “data placement scheduling” [RF03, RF03b, CCM+04], has shown fruitful results, but further investigation is needed to optimize the algorithms and to turn them into production-ready components. Significant research is also needed to develop policy-driven workflow planners and schedulers that consider space availability in a policy-cognizant manner, allocating space in a manner consistent with the priority of various workloads in a distributed, multi-virtual-organization environment.

On a more general level, a concern of replica management systems is preventing the misuse of datasets or resources, either by deliberate security violations or by inadvertent incorrect system usage. For example, the replica management system may refuse to execute a user request that would retrieve petabytes of data and monopolize distributed system resources, thereby starving other requests, or a request that would perform unauthorized or inadvertent destruction of data.

II-4.3 Data Flow between Components

Data flow between components typically is characterized by a source and destination "component" and flow(s) between the components that can be described by a transfer function. Such a function is multidimensional and, in a general sense, provides all information about how the data may be affected by the interconnecting link. Explicit data flow transfer function, quality of service profile, and other properties are a necessary part of a data-flow specification.

II-4.3.1 State of the Art

Different characteristics of data flows are more important to some classes of applications than to others. Fusion scientists need moderate transfer rates, but they need guarantees of maximum transfer time to ensure they can make appropriate adjustments to the next experiment. Remote control of instrumentation requires very low bandwidth, and scientists can work around high latency, but the latency must be constant (low jitter) or they risk damage to expensive equipment. Genomics data analyses and high-energy physics event analyses have very large aggregate bandwidths, but these flows can be distributed and are largely independent of each other. Leading-edge simulation analyses are resource intensive; and prudent computational planning, steering, and validation of the workflows (whether manual or automated) are necessary. At the beginning, flows tend to be tightly coupled and synchronous. As the scenario progresses, however, they may well be both asynchronous and more loosely coupled. Data streaming from an instrument may be irreplaceable and therefore reliability is of utmost important. Streaming video is time sequential, so retransmission is not an option; some loss of data, is allowed, but such video is also jitter sensitive.

As a rule, a scientist should not have to care how the flow occurs, so long as the performance satisfies the requirements and the flow occurs between "components" specified at an appropriate level of abstraction,. For example, the scientist should be able to state, "These data need to be at Caltech by 6AM tomorrow."

II-4.3.2 Gap Analysis and Research Needs

Although various applications each have a different set of requirements for data transport, we can distinguish broad areas where research and development is needed.

We stated that application scientists would prefer to specify data-movement constraints in a "normal" language." To achieve this goal, we need a framework that allows for specification of such parameters, with automated negotiation. Initially, we will likely need to specify data-movement-specific parameters, but in the longer term, we need schedulers and brokers that take high-level requirements and appropriately configure the data-movement services. Probably, we can build on work such as service level agreements and the WS-Agreement[20] standardization effort.

To fulfill a service level agreement for data movement requires resource management for data-movement services. Preventing unauthorized use of a data-movement system is easy, but ensuring that authorized users do not overload some part of the system is not. Resource management for data movement presents much greater challenges than does resource management of compute jobs. The vast majority of data movement uses a shared network infrastructure, making it extremely difficult to predict how long a data transfer will take. Moreover, data movement requires the coscheduling of resources at both ends of the transfer.

No single data-movement mechanism is appropriate for all data-movement tasks. A mechanism that is very efficient at moving bulk data over the wide area will likely perform poorly in an application where latency is critical. Multiresolution data flows and dynamic service coupling give rise to an even broader range of movement requirements. Transparency of service location is generally considered desirable. However, if it turns out that the source and destination are in the same process space, different mechanisms may be selected. A way of determining "locale" of source and destination, development of data-movement mechanisms for different locale cases, and transparent negotiation of these mechanisms is critical for achieving maximum performance.

Despite the exponential growth in network and hardware speeds, data-movement requirements will continue to exceed the capability of a single host. Current multistream data transfer utilities are capable of managing multiple TCP streams only between a single pair of hosts. Work is required to develop algorithms and the appropriate techniques for using multiple machines during a transfer. By design, such approaches will support activities that are resource hogs, so integration with the resource management system described above will be a requirement.

To date, the majority of data-movement work has been directed at TCP-based, packet-switched networks—the vast majority of the currently available resources. However, optical networking allows the allocation of dedicated end-to-end optical links in a rapid, automated manner. Future data-movement systems will need to take advantage of this.

[20] Web Services Agreement draft, http://www.gridforum.org/Meetings/GGF11/Documents/draft-ggf-graap-agreement.pdf

The data-movement service must be able to decide when such a circuit is appropriate, and then set up such a circuit, ideally through standard, high-level interfaces.

Dedicated circuits in turn change the rules and environment. Fairness is no longer an issue; therefore, TCP no longer must be used, and more aggressive protocols have no down side. Future data-movement systems should be engineered to be as transport-protocol-agnostic as possible, able to take maximum advantage of the environment, while still being a good network citizen.

The data-movement systems of the future will be complex and dynamic. For example, a simple service level agreement requiring data movement to be complete within 12 hours may invoke other services, allocate optical circuits, dynamically change transport protocols, and dynamically change the set of resources on which it operates during a transfer. Such systems will be "autonomic," or "self-healing," to some extent, but they will eventually still fail. Determining the source of the failure will be possible only if detailed state is exposed at every step and if appropriate troubleshooting services are available to gather this state—which out of necessity will be widely distributed—and present it in a coherent manner.

II-4.4 Multiresolution Data Movement

Multiresolution data models are those in which portions of the problem domain are represented at low resolution, while others are represented at a higher resolution. Adaptive mesh refinement (AMR) methods (see [BerkeleyAMR], [Norman1999], [Bryan2000]), allow high-resolution grids to be placed and sized precisely where needed to adequately capture physical or other detail at prescribed error tolerances. By applying refinement technique recursively, AMR supports local mesh refinement relative to the global coarse grid at scales ranging from two to six orders of magnitude, depending on the application. AMR and similar multiresolution technologies are capable of achieving resolutions levels previously impossible with a global uniform fine grid.

Multiresolution processing plays an important role in remote and distributed visualization applications, and considerable research has been done in the area of multresolution data representation and transmission. The best-known examples focus on progressive transmission of terrain-style meshes, although more recent work addresses simplification of point-sampled surfaces, edge-collapse strategies, vertex clustering, and wavelets. Reformatting datasets as multiresolution hierarchies allows a low-resolution subset of data to be sent for initial inspection, followed by progressive transmission of increasing resolution on a best-effort basis. A recent example implementation is LLNL's Terascale Browser (see [VIEWS]). The Terascale Browser uses a multiresolution data model based on a space-filling-curve storage and retrieval strategy that supports efficient access to multiple resolutions of a structured, 3-D volume. In each case, a single-resolution dataset is reorganized into a multiresolution hierarchy of components so that it can be transported and reconstructed progressively.

Unfortunately, all these techniques require significant reorganization of data in order to enable more responsive multiresolution remote visualization and data transport mechanisms. The data reorganization is expensive, but it is mandatory in order to support effective data analysis methods and interactive visual exploration. In general, it is not practical for sensors or simulation codes to output data in a multiresolution data layout suitable for analysis, visualization, and transport because these data reorganizations are extremely I/O intensive and can greatly impact the efficiency of systems designed primarily for compute-intensive workloads. Many multiresolution indexing schemes expand the data to many times the size of the original dataset. The intense I/O requirements call for criteria for system balance different from those used for running the simulation codes that produce the data. Furthermore, for practical reasons, the compute resources required to perform these data reorganizations must be placed as close as possible to the source of the data. Doing so requires significant advances in security and authorization technology in order to push data-intensive computing tasks out to the data sources.

A general-purpose infrastructure of services and reusable components that support progressive-resolution data transport requires a common data model that can fully express these hierarchical multiresolution representations. Unfortunately, no general-purpose multiresolution data model exists that can be used to build families of data analysis tools. Because “one size doesn’t fit all,” there will likely never be such a model. Efforts that assumed a top-down approach to developing common data models have repeatedly proven unworkable. It is essential that advanced data modeling efforts initiated by scientific data-management experts begin with a community-focused approach that provides data representations, data models, and components that can be reused within a well-defined domain. In the longer term, community-driven models should eventually be able to share features and components and possibly merge capabilities. Such efforts require close coordination between the scientific community and data-modeling architects using a SciDAC-like model for interdisciplinary cooperation.

II-4.5 Networking with Embedded Storage and Computation

Most approaches to managing the high volumes of data involved in data-intensive science assume the absence of any infrastructure to work with transient data. In this context, data is said to be “transient” between the time it is generated at a source (e.g., a supercomputer simulation, an instrument or detector, an aggregated set of repositories) and the time it is archived or discarded. During this time it needs to be easily available to a (potentially large and distributed) research group for analysis, visualization, or some other form of processing. These elements of the research workflow environment—the data generators, the required processing resources, and the team of people who must coordinate in the effort—are often widely spread, both geographically and administratively, across a variety of network locations. As the quantity of data involved continues to escalate, the struggle to manage transient data around this collaborative work space becomes increasingly burdensome and difficult.

At the root of the difficulty is the fact that the current Grid fabric—the combination of research networks and significant storage and computational resources attached to them—is not well adapted to the exigencies of transient management. The problems associated with buffering huge flows of data provide a prime illustration of this fact. Data inevitably needs to be buffered, for periods ranging from seconds to weeks, in order to be controlled as it moves through the distributed and collaborative research process. In order to meet the diverse and changing set of application needs of different research communities, large amounts of nonarchival storage are required for transient buffering and must be widely dispersed, easily available, and configured to maximize flexibility of use. In today's Grids, however, massive storage is concentrated mostly in data centers, available only to those with user accounts and membership in the appropriate virtual organizations, allocated as if its usage were nontransient, and encapsulated behind legacy interfaces that inhibit the flexibility of use and scheduling. This situation severely restricts the ability of some application communities to access and schedule usable storage on demand in order to make their workflow more productive.

For managing data in transit, processing resources in the current Grid are similarly constrained. The research workflow could often be made much more efficient if the data, while it is being buffered in transit, could easily be searched, reduced, reformatted, error encoded, encrypted, compressed, and so on. Even if, contrary to the current situation, significant storage resources for workflow buffering were ubiquitously deployed, having to move all the data to the edge of the network in order to perform any of these processing operations would drastically limit the efficiencies that would otherwise be gained. Consequently, effective management of transient data requires that some form of processing power, capable of being shared and scalably deployed the way network bandwidth is, be added to buffer resources as part of the common Grid fabric.

II-4.6 Security, Authorization, and Integrity

A serious problem in data management is defining and implementing security that appropriately controls access to data and data storage resources while ensuring data integrity and availability. All scientists need to be sure that their data will not be corrupted or lost, and most require security and privacy for at least some of their data. Data may be stored in a system local to the scientist, but it is more likely on a shared storage system accessible over the Internet. Access protection for data and usage controls of data resources require that storage management interoperate with an authentication system provided by the infrastructure, as well as implementing or interoperating with policy mechanisms verifying and enforcing authorization rights to access data and storage resources.

Various methods exist for authorizing a person to access a system. All methods typically require some version of a site-specific identity (such as the person's userid) is stored on the storage system. This approach does not scale easily, however, to even tens of sites and thousands of users because this information gets out of date, and it is hard to maintain and verify continuously. For this reason, the concept of a "virtual organization" (VO) is emerging to allow members access to data and storage resources based on their authenticated identity within the VO. The advantage of this approach is that only authorizations for the VO-managed identity need to be updated rather than separate authorizations for every possible site/userid that members of the VO are permitted to access. Depending on the level of access requested and the sensitivity of the data, the authorization might be performed in advance and presented offline in the form of a pass (e.g., read access for files of experiment A, ability to allocate up to 2 GB of space at a single site) while other actions might require online authorization (e.g., delete a directory of files belonging to someone else). Developing well-functioning robust VO authorization systems is still a research issue.

The VO approach helps solve the data access authorization problem for the owner/creator, but, in general, authorization is still a difficult, open problem. Consider a scientist who wishes to allow only one colleague to see a set of files. Where should this information reside? If the authorization information is managed by the storage system, then the identity of the colleague must be known and maintained by the storage system, adding significant complexity. In addition, if the set of files is replicated to other storage systems, then the authorization information needs to be propagated to these storage systems as well. On the other hand, if such information resides with the VO system, then the VO needs to manage "access control lists" for millions of files residing on multiple storage systems. Furthermore, the VO authorization manager must be notified by the file owner of any authorization changes desired. While simple VO authorization systems have been designed in the Grid community (e.g., Community Authorization Service [CAS]), no technology available today can manage authorization as part of a VO in a robust, efficient manner. Nor does the technology exist to enable authorization decisions to be communicated securely to an enforcement mechanism within the various storage management systems.

Authorization also raises the issue of allocation and enforcement of quotas. Again, a VO authorization system can be used to enforce a policy of usage by its members. It can assign usage quotas and user priority levels for various quotas sizes. For storage systems, there needs to be some measure for the quota. A decentralized approach to quotas and reservation is to define the limits of resource use for each user within the VO and to rely on a combination of self-policing and detection of abuse. The model for this approach is the Internet approach to resource sharing in communication. Self-policing, which can be enforced by the consensual use of middleware that applies appropriate resource bounds, can work in communities where most users are responsible and penalties for abuse of resources are high. Detection of abuse includes looking for patterns that violate acceptable bounds of use or periodic global auditing of resource utilization in a system too decentralized to permit continual auditing or a high degree of control. While a decentralized approach does not enable the same level of assurance of resource availability through reservation as more centralized systems, its strengths are scalability and high utilization of available resources.

In the Grid community, several efforts are under way to standardize storage allocation and usage monitoring. These include storage resource managers and NeST for various operating systems as well as network-attached storage. A complementary effort is needed for VO authorization managers, however, as well as the coordination between such systems.

Confidence about data integrity can be improved by using end-to-end techniques, analogous to those used to implement secure communication across the Internet. At the cost of application complexity and computational resources, digital signatures and other techniques that leverage secure hashes and public key encryption systems can be applied to data when it is stored, making it resilient to corruption or tampering even when replicas are made and distributed. Data integrity can be further improved by keeping data encrypted until it is under the control of an authorized reader and integrity can be verified by that reader directly, particularly for data that has internal structure.

II-4.7 Summary Table

Table II-4.1: Summary of the major issues that require further research and/or deployment efforts in the area of efficient access and queries, data integration. The ellipse marks the current position of the issue relative to pure R&D all the way on the left and off-the-shelf deployment all the way on the right.

Issues	Research and Development	Hardening and Packaging	Deployment and Maintenance	Comments
Data placement mechanisms		●		
Data placement policy	●			
Replica management		●		
Data flow between components		●		
Multiresolution data movement	●			
Networking with embedded storage		●		
Networking with embedded computation	●			
Security – authentication and authorization for data access	●			

II-5 Storage and Caching

Data storage is becoming an increasing challenge in high-performance scientific computing. In this section we focus on advances in storage technology, I/O, and automated storage techniques for meeting this challenge.

II-5.1 Storage Technology

When an application can generate 100 to 1,000 times more data, will the state of storage technology be adequate to support that data? We believe that storage capacity of disks and tapes will scale sufficiently but that high-performance access to disk and tape data will be increasingly difficult.

II-5.1.1 Magnetic Disks

For the past twelve years, disk storage capacity has scaled slightly faster than computational capability. Figure II-5.1 shows that the compound growth rate for commodity disk capacity has been over 90% per year in the past six years. Industry experts agree that capacity growth will continue but is likely to be somewhat slower.

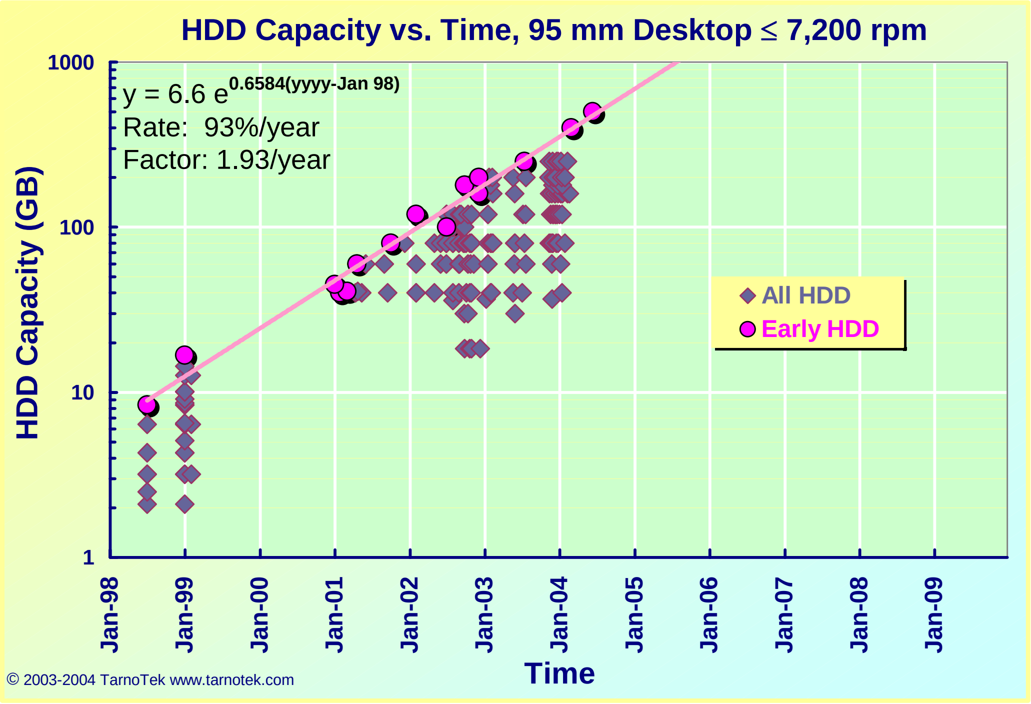


Figure II-5.1: Capacity evolution for commodity disk drives (reproduced with permission from G. Tarnopolsky, TarnoTek)

History has shown that data transfer rates to disks have *not* kept pace with computational capacity [Grochowski]. While capacity scales with areal density (track bit density times track density), transfer rates scale with linear track bit density. Transfer rate also scales with the speed of the media relative to the read/write heads, but those speeds have changed slowly; for instance, disk rotation rates have only quadrupled in over thirty years—mechanical problems grow rapidly as rotation speed rises. Figure II-5.2 shows an annual compound growth rate of about 40% for streaming transfers between the disk surface and the head assembly.

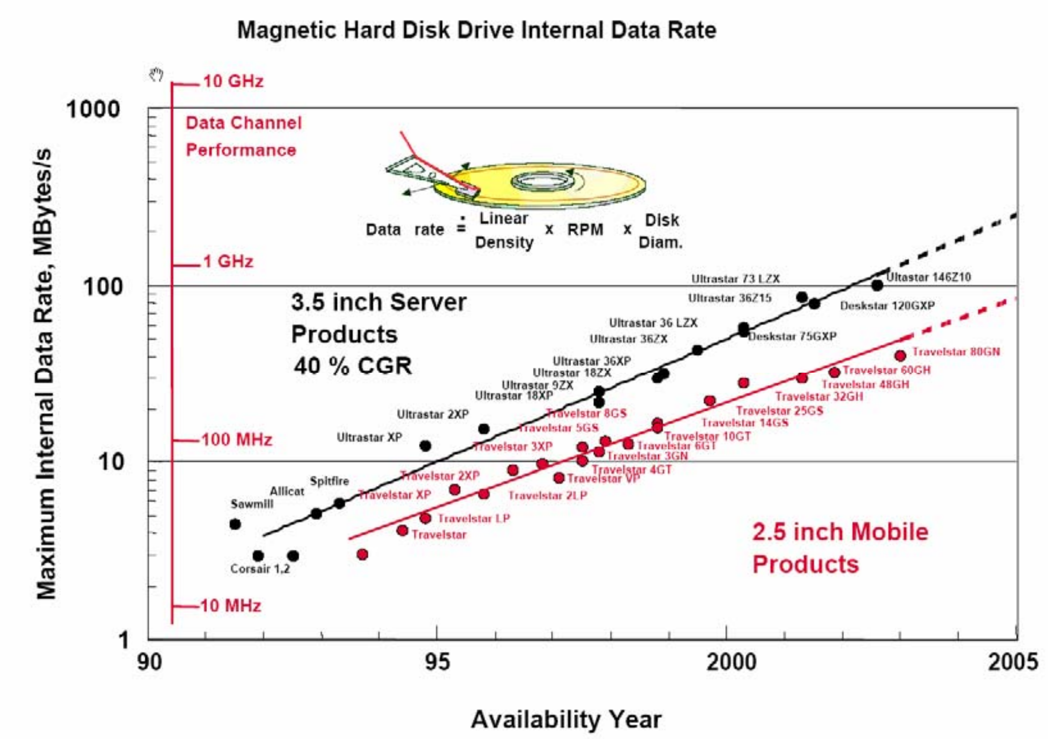
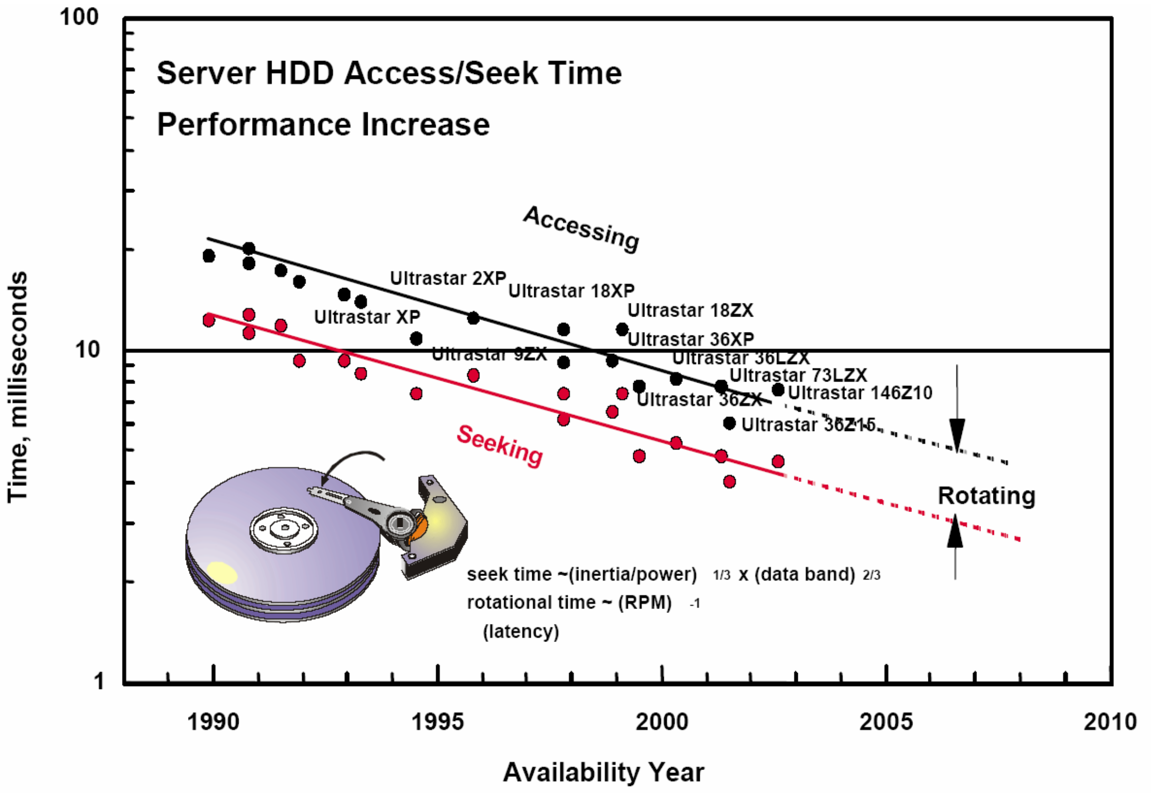


Figure II-5.2: Hard disk drive maximum internal data rate for enterprise/server mobile drives (reproduced with permission from Ed Grochowski, Hitachi Global Storage Technologies)

Rates for random access to small objects on disk are tied to rotation speed and the agility of the drive's arm movement. Figure II-5.3 shows the evolution of seek and access times for server disk drives. The trend lines in the figure correspond to an annual compound rate of decrease of less than 9%, and predictions from industry experts[21] are that the future rate of decrease will be slower.

[21] "I do not believe that there will be much shorter access times in the future," states G. Tarnopolsky, TarnoTek. "While rotation rates beyond 15K are possible in the future, these will likely occur at longer product time intervals," says Ed Grochowski, Hitachi Global Storage Technologies.



Ed Grochowski

Figure II-5.3: Seek and access times for server-class disk drives (reproduced with permission from Ed Grochowski, Hitachi Global Storage Technologies.

Incorporating disk drives into a storage system, including cache memories, I/O ordering, and RAID configurations, helps but does not solve the fundamental problems of data-transfer rates and random-access rates. Storage system performance is reported [Grochowski] to have increased by a factor of only 100 in the past thirty years, less than 17% compound annual growth.

Taking into account all factors, transfer rate has scaled significantly less rapidly than computational capacity, and random-access rates are almost stagnant. Thus, applications will be increasingly rate-limited when storing and retrieving data from magnetic disk storage devices.

The failure of transfer rate to scale fast enough portends serious problems as data quantities rise. In order for transfer rate to keep pace, more and more parallel paths will be required, which means more and more devices will be required. Random-access performance presents even more serious problems, portending the abandonment of disk technology in intense random-access applications.

II-5.1.2 Magnetic Tape

Magnetic tape continues to offer a good choice for archival storage. Error rates for tape-resident data are normally better than those for disks, and tape-resident data is less vulnerable to total data loss through device failure, accident, or malice. The areal density of recording on tape lags behind, but broadly tracks, that of recording on disk. As an archival store, tape is likely to remain five to ten times cheaper than disk and has a competitive volumetric storage density.

Tape storage becomes expensive, however, if the data must be accessed at high speed or, even worse, at high speed with an unpredictable access pattern. Buying 100 Mbytes/s streaming throughput from an array of tape drives costs 40 to 100 times as much as from a disk array. Today's robotic tape systems support efficient random access to objects of 10 gigabytes or larger but are expensive and inefficient solutions for smaller objects.

II-5.2 Parallel I/O: High-Performance Data Access for Computational Science

While many advances have been made in general-purpose parallel and cluster file systems for enterprise systems, solutions and techniques that enable end-to-end performance targeting the needs of computational science are still lacking. Near-future applications require access speeds in excess of 10 Gbytes/s. Current "hero" file I/O benchmarks are in the 1 to 10 Gbytes/s range using as many as hundreds of disks in parallel. Translating these benchmark results into comparable end-to-end I/O performance has been difficult and will become more difficult as more disks and compute processes are added to the system.

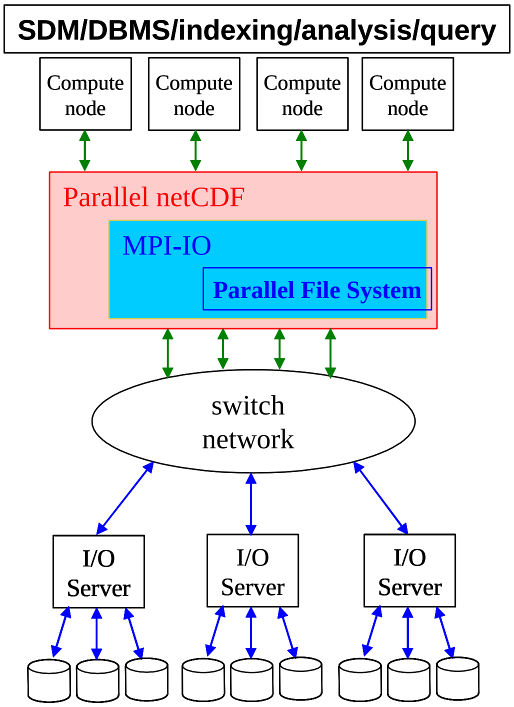


Figure II-5.4: Parallel compute servers accessing a cloud of file servers

The MPI parallel programming model is common to most of the DOE high-performance computing facilities. This model works in terms of a cloud of parallel compute processes accessing a cloud of file servers as illustrated in **Error! Reference source not found.**. In order to provide both a convenient model for access and high throughput to storage, a collection of I/O components is used. This "I/O stack" consists of three distinct layers, including high-level I/O libraries (e.g., PnetCDF and HDF5), I/O middleware (e.g., MPI-IO), and parallel file systems (e.g., PVFS2, GPFS, Lustre).

For future applications to use this I/O model, the performance of these components must be improved, particularly in the area of throughput and scalability. One way of improving the stack as a whole is to tune how components communicate with one another. For example, implementing a richer language for describing I/O accesses to the parallel file system and using this language in MPI-IO can provide significant performance gains for scientific access patterns. Also needed are interfaces between layers to facilitate the transfer of semantic information to lower levels to optimize accesses, while providing feedback to the higher layers. Furthermore, techniques that can use such interfaces to reduce synchronizations, reduce or remove need for locking, and optimize cache management will be necessary in order to scale parallel I/O functions to meet future needs. Likewise, high-level data models can better match the I/O system. The PnetCDF interface, as a replacement for HDF5, has provided as much as an order of magnitude higher I/O performance for astrophysics applications, in large part because its data model is a better match to the underlying I/O stack components.

Further work is needed to characterize workloads in the workflow models, and a clear need also exists for file systems and other layers to be enhanced and adapted to enable data streaming. Moreover, parallel I/O layers must be enhanced to provide database-like functionality, including a capability to create, manage, and use metadata along with some customizable search and query functionality that can scale to thousands of large-scale concurrent accesses.

II-5.3 Random I/O

Scientists often think of disks as ideal random-access devices; but as disk capacity increases with time, disks are become unusable for many random-access applications. If used to store thousands of 10-megabyte objects, today’s disks may still be considered to support random access to these objects—the time spent moving to the first byte of a 10-megabyte object is less than 10% of the time required for the head to read the object from the disk.[22] Many scientific applications require the retrieval of much smaller objects, often at or below the kilobyte level, resulting in retrieval rates than can be dominated by disk access time. For example, if a typical 144-gigabyte disk drive were used to store 100-byte objects, retrieving all of them in an access-time dominated mode would take over three hours—a retrieval rate of only 13 kilobytes per second, or over four thousand times slower than the disk’s streaming performance.

With a performance gap of this magnitude, caching on a modest scale cannot be expected to eliminate the problem. High-transaction-rate commercial database systems address this issue with a memory cache equal to the size of the database; waiting for disk rotation is not an option. The scale of scientific datasets makes the “cache it all in memory” approach dauntingly expensive, but in many cases a memory-dominated approach is inevitable.

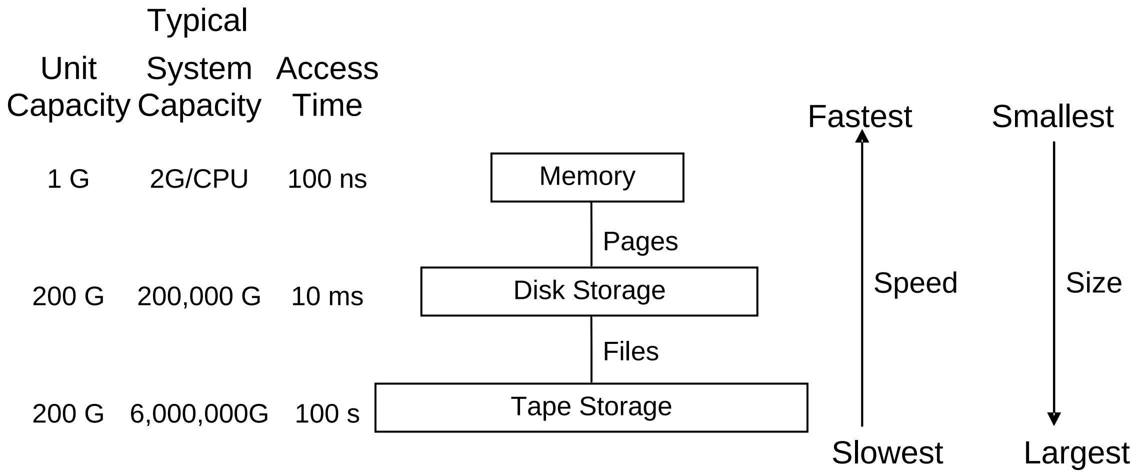
The future challenge will be to exploit large-market technologies to create in-memory scientific databases that are cost-effective. In the scientific field there has been little exploration of the memory-cache sizes needed to derandomize disk I/O. Anecdotal information shows that 1% (10 gigabytes) of data-cache memory is totally ineffective in derandomizing access to 1 terabyte of high-energy physics data. Recent proposals (e.g.,

[22] For example, a Seagate 147 gigabyte 10,000 rpm disk transfers data from the disk to the head at an average of about 60 megabytes per second and has a read access time (rotational latency plus head-seek time) of 7.7 milliseconds to deliver the first byte of an object. The access-time overhead thus falls below 10% for objects larger than 4.6 megabytes.

[Mount2004]) for approaches that are likely to be effective, call for data-cache memories of 10–100% of the data size, depending on the scientific field.

II-5.4 Dynamic Data Storage and Caching

A typical storage hierarchy for a large scientific computing center is shown in Figure II-5.5. Successive layers of the hierarchy differ in capacity by less than a factor 100 but differ in access time by factors in the range 10^4–10^5. In the case of disk versus tape, the capacity of the disk-cache layer is relatively easy to optimize, since it is at most ten times as expensive per unit capacity as the tape storage layer, and the costs become comparable if the tape system is required to support significant access throughput. In the case of memory versus disk, however, optimization is likely to be seriously skewed by the hundredfold greater cost of memory.



Unit Capacity	Typical System Capacity	Access Time
1 G	2G/CPU	100 ns
200 G	200,000 G	10 ms
200 G	6,000,000G	100 s

Figure II-5.5: Typical capacity and access time of a storage hierarchy (G=gigabytes)

Caching has long been recognized as one of the most important techniques to speed access when data is transferred between different levels of a storage hierarchy with different characteristics: speed of access, latency of access, size, and cost per bit. DRAM-based memory for caching can sometimes bridge the gap between storage system performance and compute/client requirements. Optimizing data accesses in high-performance computing requires the strategic use of several principles: overlapping of computations and I/O (whether local or remote), prestaging, caching, and data replication. Caching in this environment can be further classified into distributed, hierarchical, and adaptive.

We distinguish between caching and data replication. *Caching* involves the retention and removal of buffers to reduce access latencies to a single user or application. *Replication* implies the use of multiple copies (whole or partial) to reduce latencies in more than one user/application or for improved fault-tolerance. Caching relies on two properties of access patterns of application to be effective: *temporal locality* (if data is accessed once, it is likely to be accessed again soon) and *spatial locality* (if some data is accessed, then other data in close proximity, e.g., in RAM or on the same storage tape, is also likely to be accessed).

Large-scale storage systems have become so complex that cost of managing them can be greater than the cost of the system itself over its life-cycle. Innovative techniques are needed to address this problem. Dynamic and autonomic (and active) techniques to manage large-scale storage systems are becoming attractive, where the storage system is provided more intelligence and interfaces such that users/applications may specify what services are needed rather than how to provide them. Furthermore, quality-of-service specification and expectation will be an important part of storage systems.

II-5.4.1 State of the Art

Autonomic storage techniques are in their infancy. However, many caching techniques have been developed as systems have evolved. The quest for an optimum caching strategy has culminated in the development of numerous cache replacement policies some of which include least recently used, greedy dual size, and minimum average cost per replacement. One environment that uses caching techniques extensively in the manner envisaged for distributed data resources in scientific data management is Web-caching, where proxy servers and reverse proxy servers are configured essentially as distributed caches. Other systems that provide distributed caching functionalities are the dCache and storage resource managers.

II-5.4.2 Gaps and Needed Research

Significant work is needed in the area of availability, tuning, maintenance, and transparent fault-tolerance. While commercial offerings can support very large data stores (e.g., 1 PB), they are not designed to scale to the numbers of clients that are expected in science domains, and their failure handling is not designed for the number of I/O servers and disks necessary to provide the throughput requirements of computational science. Further, the complexity of these systems makes managing and reconfiguring in the face of failures a difficult administrative task. Moving more intelligence into the storage system can alleviate the burden placed on administrative staff, reducing the effective cost of these resources. Eventually this storage infrastructure could become autonomous, creating situations where administrative intervention is necessary only when physical devices need to be removed or added to the system. These newer directions will require a large effort in defining models, interfaces, and I/O software, including file systems, resource management, libraries, and functions.

Large-scale scientific applications need data transfers of the order of hundreds of gigabytes to tens of terabytes at very high rates among networks of data servers. Caching techniques and systems should take into account the dynamic nature as well as scales of systems and should address the following problems: latencies and data transfer rates of large data movements; collaborative caching within parallel and distributed systems as well as file-based caching; understanding and use of high-level access patterns that span datasets; and integration of caching techniques in the storage system at design time rather than adding caching to systems that are not designed to support it efficiently.

II-5.5 Summary Table

Table II-5.1 Summary of the major issues that require further research and/or deployment efforts in the area of storage and caching. The ellipse marks the current position of the issue relative to pure R&D all the way on the left and off-the-shelf deployment all the way on the right.

Issues	**Research and Development**	**Hardening and Packaging**	**Deployment and Maintenance**	**Comments**
I/O middleware for ultra-scale systems	●	●		Scalable software for portability and standardization
Parallel file systems for clusters	●	●		
Parallel file systems for ultra-scale hybrid architectures	●			To take advantage of new generation of architectures
Parallel file systems – reliability, availability, fault tolerance, and load balancing	●			Performance-reliability tradeoffs
Implementation of huge memory caches	●			Read-only caches meet most needs
Optimized cache management and distributed caching techniques	●	●		Exploitation of large-scale memories for high-performance I/O
File system and I/O support for data streaming	●			To support new data types, streaming applications and workflows
Active storage models and designs. Embedded functions, computing, software for autonomic and active storage	●			To enable computing and data analysis within active storage systems
Autonomic storage techniques. Scalable software models and implementations for autonomic storage	●			May involve changes in software interfaces and functionality between hosts and storage systems

II-6 Data Analysis, Visualization, and Integrated Environments

Next-generation science applications will present substantial challenges, for not only scientific data management, but also for scientific visualization, data analysis, and related fields. At the most basic level, the need for new capabilities in these areas is a result of the fact that computer display technologies are not advancing at anywhere near the pace of data size and complexity and the human visual system evolves only imperceptibly. Increasingly, data analysis and visualization are "where the science is done," as the raw data itself grows beyond our ability to perceive its meaning directly. Substantial effort will be required to scale data analysis and visualization tools in capability and capacity to handle the growth in data volume and complexity. Further, techniques to streamline and coordinate the use of data and visualization tools will become increasingly important.

II-6.1 Data Analysis

In order to understand datasets of increasing size and complexity, new algorithms will be required that allow researchers to quickly discover meaningful patterns without having to look directly at the raw data. Several classes of algorithms exist that have the potential to scale to the level required. However, significant research will be needed to bridge the gap between current capabilities and what will be needed for next-generation science projects. One aspect of this research will be moving the algorithms from serial to parallel implementations and porting them across various high-performance computing platforms, newer architectures, and advanced processing hardware. Many current data analysis routines are written in nonparallel languages such as IDL and are not scalable to massive datasets. While efficient parallel implementations may be trivial for some algorithms, significant redesign may be needed for others to, for example, address load imbalance issues and perhaps to move parts of the analysis closer to the data source.

II-6.1.1 Feature Identification and Tracking

Features are regions of data that satisfy some criteria, such as vortices in flow fields and flame fronts. Features can be represented directly through these criteria or indirectly through an example of the region of interest. In the latter case, there is a need for sophisticated techniques such as level sets to identify and characterize features, especially in data generated by using AMR or unstructured meshes, or on a parallel machine, where the feature may be split across processors. These algorithms will also need to be improved in terms of robustness to experimental noise and data registration issues. Tracking features as they evolve over time raises additional challenges, especially when the features split and merge over time.

II-6.1.2 Searching for Needles in Haystacks – Feature-Based and Region-Based Analysis

Detected features and regions of interest often form the basis for subsequent analysis. For example, query-based techniques can be used to retrieve flame fronts in combustion simulations, which are then analyzed to identify trends as a function of the distribution of chemical species or flow geometry. More complex queries are often formulated in terms of the raw data and require sophisticated search techniques. For example, a scientist may wish to find regions in a database that match (i.e., have a similar shape to) another plot. Such matching requires appropriate characterization of the query region and a similarity metric that can be used to retrieve regions in the dataset with similar characteristics. Work done in areas such as content-based image retrieval can be leveraged, but open research problems exist in tailoring the techniques to the needs of the scientific domain, searching in high-dimensional spaces, developing robust representations of the regions, and providing effective use of user feedback in refining the search.

II-6.1.3 Anomaly Detection in Streaming Data

Sometimes the data to be analyzed is streamed directly from an instrument or simulation (e.g., when monitoring an observation or experiment in progress). By analyzing the data as it is being generated, scientists can detect anomalies and error conditions and can steer the experiments or simulations to focus on interesting events. If the anomaly is known, a signature-based method can be used. Alternatively, the “normal” data can be modeled and deviations from that model can be flagged. Research is needed to expand existing capabilities in real-time algorithms, approximate algorithms, robust sampling techniques, and time-constrained queries, in order to handle massive and complex data.

II-6.1.4 Comparative Analysis: Verification and Validation

Scientific research often involves the comparison of two datasets, either for verification and validation or for reproducing the work of others. These comparisons are usually done at the feature level, and sometimes at the level of mesh points or pixels in an image. Research in this field, which is at its early stages, includes topics such as independent component analysis to remove characteristics specific to one simulation but not others and the development of metrics for comparison at the feature level.

II-6.1.5 Data Fusion and Link Analysis

Many application domains mine information across disparate sources of data such as journal papers, Web sites, and data from complementary experiments. Data fusion techniques are needed to combine these data sources so they can be analyzed as one. Significant work has been done in the context of Web mining and intelligence analysis in developing methods to analyze hypertext links between data such as text, documents, and figures. Progress has also been made in techniques to cluster and infer links between unstructured text documents, However, significant effort will be required to generalize these techniques to scientific data, including the use of semantic information in link graph analysis and the ability to include domain-specific knowledge and techniques to appropriately represent different types of data within more general clustering and link analysis models.

II-6.2 Visualization

Scientific visualization is the transformation of abstract data into images that are more readily comprehensible than the data itself. It is the primary means by which scientists "see" their data, and it forms a central part of most, if not all, scientific processes. Visualization techniques can be applied to raw data as well as to the results of analyses; and, to a large degree, the challenges raised by increased data size and complexity for visualization mirror those of analysis tools. New visualization techniques are needed to make the patterns in large, complex datasets stand out. Visualization algorithms applied to raw data and, increasingly, to large subsets derived by analyses will need to shift from serial to parallel designs, including parallel transfer of data to large displays.

II-6.2.1 Visual Representation of Multidimensional, Multimodal Data

Most visualization research to date has focused on algorithms for static, single-variable fields, with a few efforts producing results for simple vector field visualization. However, data is increasingly being captured as a function of many dimensions (up to hundreds) and nonuniform coordinates. Advances in simulations that focus on physics, for example, produce highly complex output that defies analysis with current visualization technology. Another example is the angular distribution of radiation computed in radiation transport codes used in modeling supernova explosions. Other examples stem from computational biology and combustion, where chemical pathways are ill understood because of their complexity. In order to support next-generation science, advances will be required in multidimensional visualization, visualization of time-varying data, and methods for visualizing information that may have no natural mapping to space-time coordinates.

II-6.2.2 Visual Comparative Analysis

As the number and complexity of datasets grows, quick visual comparative analysis will become increasingly important. Such visualization might involve direct display of pixel-level differences between datasets, visual display of statistics about the differences, or more domain-specific displays depicting, for example, areas of similarity/conservation in the genetic sequences of multiple species. As new comparative analysis techniques are developed, new visual representations will be required. For example, combining derived information such as multidimensional statistical information and uncertainty, or feature information, with an underlying data visualization would provide significant new capabilities for understanding the differences in data produced by different simulation runs or experiments.

II-6.2.3 Interactive Visual Data Exploration

Data visualizations are also used as a means to explore data (i.e., with navigation through the visual space selecting regions of interest). Interactive visual exploration has proven to be a powerful tool, allowing researchers to concentrate on science rather than the mechanics of interacting. In order to apply these techniques to large, multidimensional datasets, a range of advances will be required. Interaction with large datasets will require very efficient parallel data pipelines from source to construct three-dimensional displays, and animation will be required to allow additional data dimensions to be represented. Furthermore, collaborative visualization—which will stress distributed computing, data management, and networking infrastructure as data volume increases—will become increasingly useful as the range of phenomena within single datasets increases. Multiresolution techniques, allowing researchers to efficiently zoom to focus on patterns at different scales, are another key aspect of interactive visualization. Early efforts in multiresolution visualization include the ChomboVis application, tailored to display data from AMR grids, and the Terascale Browser, which uses a custom multiresolution representation based on a space-filling curve to provide efficient access to large data at varying resolutions. Research is needed to produce standard multiresolution data models and efficient general algorithms that can be used across scientific disciplines.

II-6.3 Integrated Environments

The term *integrated data analysis and visualization environment* (IDAVE) refers to an integrated and unified set of software tools that provide an end-to-end solution for analysis and visualization of scientific data and simulation results. IDAVEs can accelerate the process of scientific inquiry and discovery and can lower the barriers to incorporating new techniques and performing research across disciplines. The central challenges in realizing IDAVEs relate to the fact that, while the value of IDAVEs grow rapidly with their scope (i.e., with the fraction of daily work a researcher can do within the environment), integration costs also grow rapidly. Thus, stable standard data models form the core of IDAVE implementations. Choices made about the data model directly affect the tradeoff between depth of integration and extensibility.

IDAVEs support one or more interaction modalities. In “vertical integration,” the constituent technologies—data models, software components, workflow management, and so forth—are combined into something analogous to a finished application. In “horizontal integration,” similar technologies are combined from different sources, such as federating databases, so that they that appear as a single large data cache (often delivered as programming libraries). Combining these two modalities are toolkit approaches in which researchers select and script the constituent technologies to customize the toolkit to solve a specific task. Spanning this range from graphical application-like suites to lower-level frameworks that simplify construction of data and workflows, IDAVEs have been highly successful in increasing productivity and enabling software reuse. While opinions differ within the scientific applications community about which style of IDAVE is most helpful, IDAVEs of some form clearly will become increasingly critical for researcher productivity and effective software reuse. Producing next-generation IDAVEs will require end-to-end coordination across the proposed program, specifically with respect to models for managing data flow and appropriate programming and visual abstractions for representing data management processes.

II-6.3.1 State of the Art

A number of well-known visualization applications from both commercial ([AVS], AVS/Express, Khoros, [OpenDX]) and research organizations (apE,[23] SCIrun) implement a toolkit-style visual programming interface to a data-flow-based execution model. Researchers draw lines that represent typed data flows between software modules that

[23] The Animation Production Environment, or apE, originated from the Ohio State University and has since been transferred to Taravisual Corporation, 929 Harrison Avenue, Columbus, OH 43215.

perform atomic operations on data objects (e.g., specifying the extraction of a two-dimensional slice from a three-dimensional array, which is the passed to an image viewer module). Strong data typing and the visual programming model have proven straightforward for nonexperts to learn and use, but use becomes more difficult as the number of data types and complexity of data processing increase. When data or module parameters change, the data flow network executes. In most cases, these environments are limited to execution on a single machine and cannot, for example, realize visual networks as grid workflows. Object-oriented (OO) environments for data analysis and visualization have similarly originated from both commercial (VTK) and research organizations. These environments consist of both class libraries [ROOT] and complete applications [Ecce] built around well-defined data models and atomic operations on data objects defined in the classes. Developer-level expertise is required to create or modify applications in an OO environment, but the level of integration can be much greater than is common in toolkit approaches. Like data-flow and OO environments, interpreted-language environments originate both in industry [IDL] and in research [CDAT].[24] The interpreted-language environments provide a high-level programming language for data manipulation, analysis, and visualization capabilities that are accessed via "subroutine" calls from an interpreted language front-end. The interpreted language supports common language constructs such as loops, subroutines, and conditionals as well as higher-level constructs applied to complex data types (e.g., matrix multiplication). These systems are also extensible in terms of their programmability and their ability to link with other software components. The ease with which new, external software may be used in these environments varies from implementation to implementation.

II-6.3.2 IDAVEs: Gaps and Research Needed

At the core of the IDAVE concept is the notion that software components in the environment share concepts related to data flow and data types. In order to function in the context of the proposed advanced data management technologies, IDAVEs will require a model that simultaneously incorporates concepts such as remote execution, data provenance, data integration services, and data replicas. Standardization of data primitives and mechanisms for invoking data flow services and recording provenance information, for example, would be one mechanism that woud provide the necessary level of coordination. Alternatively, or in addition, the incorporation of semantic data description and data integration capabilities could support mechanisms to automate aspects of data conversion and the integration of new algorithms and services.

IDAVEs will require the development of high-level abstractions for working with data management services to, for example, allow researchers to compose a data analysis and visualization pipeline involving distributed data sources and parallel algorithms as easily desktop pipelines can be composed today. Programmatic and graphical representations of data sources, data types, and data management services will be needed. In order to

[24] Over time, CDAT has evolved from being a purely interpreted-language interface to include a visual front end that invokes the functions previously accessible only from a Python script.

accommodate different levels of access to the underlying complexity of the data processing workflow, multiple levels of abstraction will be needed. Mechanisms for exposing details within a toolkit approach and then hiding them to produce application-like functionality will have to be developed. While some conceptual models exist for such rich IDAVE interfaces, significant work will be needed to realize systems capable of supporting activities, with dynamically varying levels of detail, across the planning, execution, and exploration phases of research.

While IDAVE development is likely to remain an area of active research for some time, incremental steps should be taken within the proposed program. The data-management capabilities discussed throughout this document are clearly necessary to support next-generation science, but using them effectively could require significant expertise and effort on the part of practicing researchers. IDAVE data and process models, coupled with integrating programming and graphical interfaces, will simplify common tasks, automate the mechanics of using advanced data-management technologies, and enable reuse of analysis, visualization, and other technologies.

II-6.4 Summary Table

Table II-6.1 Summary of the major issues that require further research and/or deployment efforts in the area of storage and caching. The ellipse marks the current position of the issue relative to pure R&D all the way on the left and off-the-shelf deployment all the way on the right.

Issues	Research and Development	Hardening and Packaging	Deployment and Maintenance	Comments
Feature extraction and tracking	⬭			Algorithms for non-Cartesian meshes; sophisticated tracking algorithms
Searching for needles in hay-stacks	⬭			Algorithms to support massive datasets
Anomaly detection in streaming data	⬭			Need approximate, real-time, 1-pass algorithms
Comparative analysis and visualization	⬭			Need features and metrics for comparison
Data fusion and link analysis	⬭			Need algorithms to fuse multimodal data and find associations among them
Scalable data analysis	⬭			Algorithms for parallel distributed data; need to address load balancing
Advanced visualization techniques	⬭			Feature-based, multiresolution, multimodal
Interactive visual data exploration	⬭			
Standardized process and data models for data flow	⬭			
IDAVE architecture and design	⬭			
IDAVE interface development	⬭			

References

[Aalst00] W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros, “Advanced Workflow Pattern,” in 7th International Conference on Cooperative Information Systems (CoopIS 2000), vol. 1901 of *Lecture Notes in Computer Science*, edited by O. Etzion and P. Scheuermann, Springer-Verlag, Berlin, 2000, p. 18. http://tmitwww.tm.tue.nl/research/patterns/download/coopis.pdf

[Aldering2002] G. Aldering et al. (+ 72 authors), “Overview of the SuperNova/Acceleration Probe (SNAP),” in *Future Research Direction and Visions for Astronomy*, Proceedings of the SPIE, vol. 4835, edited by Alan M. Dressler, 2002, pp. 146–157.

[Atkinson 2003] M. Atkinson, Ann L. Chervenak, Peter Kunszt, Inderpal Narang, Norman W. Paton, Dave Pearson, Arie Shashoni, and Paul Watson, “Chapter 22: Data Access, Integration and Management,” in *The Grid: Blueprint for a New Computing Infrastructure, Second Edition*, edited by I. Foster and C. Kesselman, Morgan Kaufmann, 2003.

[AVS] http://www.avs.com

[Bent2002] John Bent, Venkateshwaran Venkataramani, Nick LeRoy, Alain Roy, Joseph Stanley, Andrea C. Arpaci Dusseau, and Remzi H. Arpaci, “Flexability, Manageability, and Performance,” in *Grid Storage Appliance, Proceedings of the HPDC*, 2002, pp. 3–12.

[BerkeleyAMR] http://seesar.lbl.gov/AMR/Overview/index.html

[Blondin2003] J. M. Blondin, A. Mezzacappa, and C. DeMarino, “Stability of Standing Accretion Shocks, with an Eye toward Core Collapse Supernovae” *Astropysics Journal* 584, 2003, pp. 971–980.

[Bryan2000] G. L. Bryan, “Fluid in the Universe: Adaptive Mesh Refinement in Cosmology,” *Computing in Science and Engineering* 1, no. 2, March/April 2000, pp. 46–53.

[Cai2004] M. Cai, A. Chervenak, and M. Frank, “A Peer-to-Peer Replica Location Service,” in *Proceedings of SC2004 Conference,* November 2004 (to appear).

[CAS] http://www.globus.org/security/CAS/GT3

[CCM+04] David G. Cameron, Ruben Carvajal-Schiaffino, A. Paul Millar, Caitriana Nicholson, Kurt Stockinger, and Floriano Zini, “Analysis of Scheduling and Replica Optimisation Strategies for Data Grids Using OptorSim,” *International Journal of Grid Computing* (to appear).

[CDAT] http://est.llnl.gov/cdat/.

[CDF+02] A. Chervenak, E. Deelman, I. Foster, L. Guy, W. Hoschek, A. Iamnitchi, C. Kesselman, P. Kunst, M. Ripeanu, B, Schwartzkopf, H, Stockinger, K. Stockinger, and B. Tierney, “Giggle: A Framework for Constructing Scalable Replica Location Services,” in *Proceedings of Supercomputing 2002 (SC2002)*, November 2002.

[Chervenak 2004] A. L. Chervenak, Naveen Palavalli, Shishir Bharathi, Carl Kesselman, and Robert Schwartzkopf, "Performance and Scalability of a Replica Location Service," presented at the High Performance Distributed Computing Conference (HPDC-13), Honolulu, June 2004.

[Chou00] A. Choudhary, M. Kandemir, J. No, G. Memik, X. Shen, W. Liao, H. Nagesh, S. More, V. Taylor, R. Thakur, and R. Stevens, "Data Management for Large-Scale Scientific Computations in High Performance Distributed Systems,'" *Cluster Computing: The Journal of Networks, Software Tools and Applications* 3, no. 1, 2000, pp. 45–60.

[Deelman2002] E. Deelman, J. Blythe, Y. Gil, and C. Kesselman, "Pegasus: Planning for Execution in Grids," GriPhyN Project Technical Report 2002-20, 2002.

[Domenico2002] B. Domenico, J. Caron, E. Davis, R. Kambic, and S. Nativi., Thematic Real-Time Environmental Distributed Data Services (THREDDS), *Journal of Digital Information* 2, no. 4, article no. 114, 2002-05-29.

[ecce] http://ecce.emsl.pnl.gov.

[GridFTP2004] Globus Alliance, "The GridFTP Protocol and Software," http://www.globus.org/datagrid/gridftp.html.

Grochowski] E. Grochowski and R. D. Halem, "Technological Impact of Magnetic Hard Disk Drives on Storage Systems," *IBM Systems Journal*, 42, no. 2, 2003, p. 338.

[IDL] http://www.rsinc.com/idl/.penD

[Kepler] Kepler, "A System for Scientific Workflows," http://kepler.ecoinformatics.org

[Kosar2004] Tevfik Kosar and Miron Livny, "Stork: Making Data Placement a First Class Citizen in the Grid," in *Proceedings of 24th IEEE Int. Conference on Distributed Computing Systems (ICDCS2004)*, Tokyo, March 2004 (to appear).

[KLSS03] Peter Kunszt, Erwin Laure, Heinz Stockinger, and Kurt Stockinger, "*Advanced Replica Management with Reptor*," in *Proceedings of the International Conference on Parallel Processing and Applied Mathematics*, Springer-Verlag, 2003.

[Ma04] X. Ma, M. Winslett, J. Norris, X. Jiao, and R. Fiedler, "GODIVA: Lightweight Data Management for Scientific Visualization," 20th International Conference on Data Engineering (ICDE), 2004

[Mount2004] R. Mount, "A Leadership-Class Facility for Data-Intensive Science", http://www-user.slac.stanford.edu/rmount/leadership/HighEndComputingProposal--4_9_04.doc

[No03] Jaechun No, Rajeev Thakur, and Alok Choudhary, "High-Performance Scientific Data Management System," *Journal of Parallel and Distributed Computing* 63, no. 4, 2003, pp. 434–447.

[Norman1999] M. L. Norman, J. Shalf, S. Levy, and G. Daues, "Data Management and Visualization Strategies for Adaptive Mesh Refinement Simulations," *Computing in Science and Engineering*, July/August. 1999, pp. 22–32.

[OpenDX] http://www.opendx.org

[PADB0303] J. S. Plank, S. Atchley, Y. Ding, and M. Beck, "Algorithms for High Performance, Wide-Area, Distributed File Downloads," *Parallel Processing Letters* 13, no. 2, June 2003, pp. 207–224.

[RF02] Kavitha Ranganathan and Ian Foster, "Decoupling Computation and Data Scheduling in Distributed Data-Intensive Applications," in *Proceedings of the IEEE International Symposium for High Performance Distributed Computing (HPDC-11),* August 2002, p. 352.

[RF03] Kavitha Ranganathan and Ian Foster, "Simulation Studies of Computation and Data Scheduling Algorithms for Data Grids," *Journal of Grid Computing* 1, no. 1, 2003, pp. 53–62.

[RF03b] Kavitha Ranganathan and Ian Foster, "Computation Scheduling and Data Replication Algorithms for Data Grids, in *Grid Resource Management*, edited by J. Weglarz, J. Nabrzyski, J. Schopf, and M. Stroinski, Kluwer, 2003.

[RFT2004] "Reliable File Transfer Service,"

http://www-unix.globus.org/ogsa/docs/alpha3/services/reliable_transfer.html.

[ROOT] http://root.cern.ch.

[SDB] Scientific Data Browser. http://hdf.ncsa.uiuc.edu/sdb/sdb.html

[Singh2003] G. Singh, S. Bharathi, A. Chervenak, E. Deelman, C. Kesselman, M. Manohar, S. Patil, and L. Pearlman, "A Metadata Catalog Service for Data Intensive Applications," in *Proceedings of SC03*, Phoenix, Arizona, November 2003, http://www.globus.org/research/papers/mcs_sc2003.pdf.

[SRB] http://www.npaci.edu/DICE/SRB/.

[SSS04] Arie Shoshani, Alex Sim, and Kurt Stockinger," Replica Management Component Services - Functional Interface Specification," work in progress, Berkeley, CA, 2004.

[Tyson2002] J. A. Tyson, "Large Synoptic Survey Telescope: Overview," in *Survey and Other Telescope Technologies and Discoveries*, *Proceedings of the SPIE,* vol. 4836, edited by J. Anthony Tyson and SidneyWolff, 2002, pp. 10–20.

[VIEWS] http://www.llnl.gov/icc/sdd/img/terascale_views.shtml

Organizing Committee

Bair	Raymond	ANL	bair@mcs.anl.gov
Becla	Jacek	SLAC	becla@slac.stanford.edu
Bethel	Wes	LBNL	ewbethel@lbl.gov
Choudhary	Alok	Northwestern University	choudhar@ece.northwestern.edu
Diachin	Lori	LLNL	Diachin2@LLNL.GOV
Drake	John	ORNL	drakejb@ornl.gov
Gaines	Irwin	Fermilab/DOE	gaines@fnal.gov
Hanushevsky	Andrew	SLAC	abh@stanford.edu
Kent	Stephen	Fermilab	skent@fnal.gov
Klasky	Scott	PPPL	sklasky@pppl.gov
Mezzacappa	Anthony	ORNL	mezzacappaa@ornl.gov
Michaels	George	PNNL	george.michaels@pnl.gov
Moore	Reagan	SDSC	moore@sdsc.edu
Mount	Richard	SLAC	richard.mount@stanford.edu
Olson	Doug	LBNL	dlolson@lbl.gov
Petravick	Donald	Fermilab	petravick@fnal.gov
Pordes	Ruth	Fermilab	ruth@fnal.gov
Schissel	David	General Atomics	schissel@fusion.gat.com
Shelton	William	ORNL	sheltonwajr@ornl.gov
Shoshani	Arie	LBNL	ashoshani@lbl.gov
Stevens	Rick	ANL/University of Chicago	stevens@mcs.anl.gov
van Rosendale	John	DOE	JohnVR@er.doe.gov
Weeks	William	SLAC	wcw@slac.stanford.edu
Williams	Dean	LLNL	williams13@llnl.gov

Workshop Participants

Ambrosiano	John	LANL	ambro@lanl.gov
Anderson	Ian	ORNL	andersonian@sns.gov
Anderson	Gordon	PNNL	gordon@pnl.gov
Atchley	Scott	University of Tennessee	atchley@cs.utk.edu
Babnigg	Gyorgy	ANL	gbabnigg@anl.gov
Bader	Dave	LLNL	bader2@llnl.gov
Bair	Raymond	ANL	bair@mcs.anl.gov
Bauerdick	Lothar	Fermilab	bauerdick@fnal.gov
Beck	Micah	University of Tennessee	mbeck@cs.utk.edu
Becla	Jacek	SLAC	becla@slac.stanford.edu
Bethel	Wes	LBNL	ewbethel@lbl.gov
Blondin	John	North Carolina State University	john_blondin@ncsu.edu
Bunn	Julian	Caltech	Julian.Bunn@caltech.edu
Chen	Jacqueline	Sandia National Laboratories	jhchen@sandia.gov
Chervenak	Ann	USC Information Sciences Institute	annc@isi.edu
Chin	George	PNNL	george.chin@pnl.gov
Choudhary	Alok	Northwestern University	choudhar@ece.northwestern.edu
Cook	Kem	LLNL	kcook@llnl.gov
Coverston	Harriet	Sun Microsystems	harriet.coverston@sun.com
Cowles	Bob	SLAC	bob.cowles@stanford.edu
Critchlow	Terence	LLNL	critchlow@llnl.gov
Dee	Richard	StorageTek	richard_dee@storagetek.com
Diachin	Lori	LLNL	Diachin2@LLNL.GOV
Ding	Chris	LBNL	chqding@lbl.gov

Doyle	Michael	Eolas Technologies Inc.	mike@eolas.com
Drake	John	ORNL	drakejb@ornl.gov
Elbert	Stephen	IBM	selbert@us.ibm.com
Gaeta	Michael	LANL	mgaeta@lanl.gov
Gaines	Irwin	Fermilab/DOE	gaines@fnal.gov
Gibbard	Bruce	BNL	gibbard@bnl.gov
Gibbons	Lawrence	Cornell	lkg@mail.lepp.cornell.edu
Gibson	Garth	Panasas	garth.gibson@panasas.com
Giometti	Carol	ANL	csgiometti@anl.gov
Golden	David	Stanford University	david.golden@stanford.edu
Gray	Jim	Microsoft	Gray@Microsoft.com
Guha	Aloke	COPAN Systems	aloke.guha@copansys.com
Guzenda	Leon	Objectivity, Inc.	leon@objy.com
Hadig	Thomas	SLAC	hadig@slac.stanford.edu
Hanushevsky	Andrew	SLAC	abh@stanford.edu
Herwig	Kenneth	ORNL	herwigkw@ornl.gov
Hulen	Harry	Representing IBM HPSS Project	hulen@us.ibm.com
Jacob	Robert	ANL	jacob@mcs.anl.gov
Jacobsen	Janet	U.C. Berkeley/LBNL	jsjacobsen@lbl.gov
Johnston	William	LBNL	wej@es.net
Jones	Philip	LANL	pwjones@lanl.gov
Kamath	Chandrika	LLNL	kamath2@llnl.gov
Karp	Peter	SRI International	pkarp@ai.sri.com
Kennedy	Robert	Fermilab	kennedy@fnal.gov
Kent	Stephen	Fermilab	skent@fnal.gov
Kerschberg	Larry	George Mason University	kersch@gmu.edu
Klasky	Scott	PPPL	sklasky@pppl.gov
Kohl	James	ORNL	kohlja@ornl.gov

Kruger	Scott	Tech-X Corporation	kruger@txcorp.com
Layton	Will	COPAN systems	will@copansys.com
Lee	Wei-Li	PPPL	wwlee@pppl.gov
Livny	Miron	University of Wisconsin-Madison	miron@cs.wisc.edu
Louis	Steve	LLNL	stlouis@llnl.gov
Luitz	Steffen	SLAC	luitz@slac.stanford.edu
Malon	David	ANL	malon@anl.gov
Maltsev	Natalia	ANL	maltsev@mcs.anl.gov
Marshall	Stuart	KIPAC/SLAC	marshall@slac.stanford.edu
Matarazzo	Celeste	LLNL	matarazzo1@llnl.gov
McPhillips	Timothy	SLAC	tim@slac.stanford.edu
Merritt	K. Wyatt	Fermilab	wyatt@fnal.gov
Meza	Juan	LBNL	jcmeza@lbl.gov
Mezzacappa	Anthony	ORNL	mezzacappaa@ornl.gov
Michaels	George	PNN L	george.michaels@pnl.gov
Miller	Ethan	University of California, Santa Cruz	elm@cs.ucsc.edu
Miller	Steve	ORNL - SNS	millersd@ornl.gov
Moore	Reagan	SDSC	moore@sdsc.edu
Moore	Terry	Univ. of Tennessee	tmoore@cs.utk.edu
Mount	Richard	SLAC	richard.mount@stanford.edu
Myers	James	PNNL	Jim.Myers@pnl.gov
Myra	Eric	State University of New York at Stony Brook	emyra@mail.astro.sunysb.edu
Najm	Habib	Sandia National Laboratories	hnnajm@sandia.gov
Naughton	Jeff	University of Wisconsin	naughton@cs.wisc.edu

Needham	Shawn	University of Chicago, ASCI Flash Center	shawn@flash.uchicago.edu
Newman	Henry	Instrumental Inc	hsn@instrumental.com
Nishtala	Satya	Sun Microsystems	satya@sun.com
Nugent	Peter	LBNL	penugent@LBL.gov
O'Keefe	Michael	LBNL	maok@lbl.gov
Oldfield	Ron	Sandia National Laboratories	raoldfi@sandia.gov
Olson	Doug	LBNL	dlolson@lbl.gov
Otoo	Ekow	LBNL	ekw@data.lbl.gov
Parker	Steven	University of Utah	sparker@cs.utah.edu
Pascucci	Valerio	LLNL	pascucci@llnl.gov
Perl	Joseph	SLAC	perl@slac.stanford.edu
Petravick	Donald	Fermilab	petravick@fnal.gov
Philpott	Sandy	Jefferson Lab	Sandy.Philpott@jlab.org
Plewa	Tomasz	The ASCI Flash Center, The University of Chicago	tomek@flash.uchicago.edu
Pordes	Ruth	Fermilab	ruth@fnal.gov
Pouchard	Line	ORNL	Pouchardlc@ornl.gov
Rahn	Larry	Sandia National Laboratories	rahn@sandia.gov
Rao	Nageswara	ORNL	raons@ornl.gov
Riccardi	Greg	Florida State University	riccardi@cs.fsu.edu
Riedel	Richard	ORNL	riedelra@sns.gov
Romano	Raquel	LBNL	raromano@lbl.gov
Rosasco	Gregory	National Institute of Standards and Technology	gregory.rosasco@nist.gov
Ross	Rob	ANL	rross@mcs.anl.gov
Rotem	Doron	LBNL	d_rotem@lbl.gov

Samatova	Nagiza	ORNL	samatovan@ornl.gov
Samet	Hanan	University of Maryland	hjs@cs.umd.edu
Schissel	David	General Atomics	schissel@fusion.gat.com
Scott	Mary Anne	Dept of Energy	scott@er.doe.gov
Shasharina	Svetlana	Tech-X Corporation, University of Colorado	sveta@txcorp.com
Shelton	William	ORNL	sheltonwajr@ornl.gov
Shoshani	Arie	LBNL	ashoshani@lbl.gov
Smith	Tim	CERN	Tim.Smith@cern.ch
Smith	Todd	Geospiza	todd@geospiza.com
Steenberg	Conrad	Caltech	conrad@hep.caltech.edu
Stevens	Rick	ANL/University of Chicago	stevens@mcs.anl.gov
Stockinger	Kurt	LBNL	KStockinger@lbl.gov
Straatsma	TP	PNNL	tps@pnl.gov
Strand	Gary	NCAR	strandwg@ucar.edu
Studham	Scott	PNNL	scott.studham@pnl.gov
Swesty	Doug	SUNY @ Stony Brook	dswesty@mail.astro.sunysb.edu
Tarnopolsky	Giora	INSIC/Tarnotek	gjtarno@tarnotek.com
Thakur	Rajeev	ANL	thakur@mcs.anl.gov
Tideman	Sonja	Sandia National Laboratory	stidema@sandia.gov
Trunov	Artem	SLAC	artem@slac.stanford.edu
van Lingen	Frank	Caltech	fvlingen@caltech.edu
van Rosendale	John	US Dept. of Energy	JohnVR@er.doe.gov
Vouk	Mladen	North Carolina State University	vouk@ncsu.edu
Wang	Nanbor	Tech-X	nanbor@txcorp.com

		Corporation	
Weeks	William	SLAC	wcw@slac.stanford.edu
Wehner	Michael	LBNL	mfwehner@lbl.gov
Wenaus	Torre	BNL	torre@wenaus.com
Westhead	Martin	EPCC, University of Edinburgh	M.Westhead@epcc.ed.ac.uk
Whitney	Alan	MIT Haystack Observatory	awhitney@haystack.mit.edu
Wilde	Michael	ANL	wilde@mcs.anl.gov
Wiley	H Steven	PNNL	steven.wiley@pnl.gov
Winslett	Marianne	University of Illinois	winslett@cs.uiuc.edu
Woodward	Paul	University of Minnesota	paul@lcse.umn.edu
Wu	Kesheng	Berkeley Lab	John.Wu@ACM.org
Wuerthwein	Frank	UCSD	fkw@ucsd.edu
Young	Charles	SLAC	young@slac.stanford.edu