Data Access and Analysis of Massive Datasets for High-Energy and Nuclear Physics

Participants:

LBNL		
	NP	D. Olson (PI), G. Odyniec, I. Sakrejda, J. Marx
	HEP	J. Siegrist (PI), I. Hinchliffe, R. Jacobsen
	Computing	D. Quarrie, W. Johnston, B. Tierney, N. Johnston, S. Lau, K. Campbell, A. Shoshani, D. Rotem, W. Barber, H. Simon
ANL		E. May, D. Malon
BNL		B. Gibbard
FSU		G. Riccardi, L. Dennis
UCLA		H. Huang
U.Tenn.		S. Sorensen
Yale		S. Kumar

1. Introduction

Advances in computational capabilities, information management, and multi-user data access are essential if the next generation of experiments in both high energy and nuclear physics are to be able to fully address the forefront scientific issues for which they are designed. Among these forefront issues are two most fundamental questions facing high energy and nuclear physics today, namely characterization of the transition to the Quark-Gluon Plasma (QGP) phase of matter and the discovery of the mechanism responsible for electro-weak symmetry breaking[1].

These experiments will record and analyze data from physics events of unprecedented complexity. The resulting data streams of up to tens of megabytes per second and the requirements for "data mining" in huge (tens of terabytes) data sets by multiple, geographically distributed teams of scientists set the scale of this Grand Challenge proposal. Simple extrapolations of existing techniques will not be sufficient; new approaches and solutions are required. The Energy Research Supercomputer Centers have a major role in both addressing this intellectual challenge and subsequently in the implementation of solutions that complement the onsite computational capabilities envisioned for the host laboratories of the experiments.

This Grand Challenge Application (GCA) proposal is the result of an unprecedented collective effort of physicists who are participating in the next generation of major experiments supported by DOE's HENP Program (STAR and PHENIX for RHIC, ATLAS for LHC, BABAR for the SLAC B-factory and CLAS for CEBAF)[2] and by computer scientists who will contribute to the solutions needed to address this challenge. The resulting multi-institutional team seeks a common solution because of the similarity of the problems faced by these experiments. Solutions to this "Grand Challenge" will permit major advances in capability for both the high energy and nuclear physics programs of DOE's Office of Energy Research.

The coming generation of HENP experiments will produce orders of magnitude more data than their predecessors; tens to hundreds of terabytes (TB) of raw data per year for each experiment

and equivalent amounts of Monte Carlo simulated data to model detector response, detector acceptance and provide a baseline for looking for "new physics." Even after the first level of analysis, each experiment will be left with many tens of TB of data per year. The data must be available to a hundred or more collaborators per experiment, spread across the United States and the world. Effective analysis of this reduced data requires that it be accessed multiple times. The total cpu power required is several 100 GFLOPS for the larger experiments. Then, the tens of TB of processed data must be sorted and further analyzed by many researchers in order to extract publishable scientific results on a wide variety of topics.

Three challenging aspects of data management and access must be solved: 1) efficient organization of the data to be stored, managed, and accessed to allow timely selection of interesting events (avoid full data set reads); 2) providing the cpu cycles and massive parallelism required for analysis and simulations; and 3) development of the software and remote access environment that permits many (100) physicists to individually select the data sets of interest and implement particular physics analysis algorithms.

This Grand Challenge project will demonstrate a system with capability for physics analyses which require up to 100 Gflops of parallel computing capacity and data sets with sizes of up to 50 TB that have passed through a first level of analysis to enrich the information density by an order of magnitude. In addition, the system will demonstrate the capability to generate appropriate simulated data for analyses and provide access to data subsets from remote sites for software development, event visualization and physics analysis.

The functional goals suit the requirements of all of the participating experiments and the scale of demonstrations are the size of any one except ATLAS, for which it serves as a large scale prototype. We will develop a portable solution which can be used by a number of experiments simultaneously and installed at other computing centers. We will demonstrate the use of this system for several nuclear physics experiments and prototype the use for a number of high energy experiments. This implementation will validate the experiment analysis system interface design to ensure applicability to other large HENP experiments. Portability will be demonstrated by installing and running the system at NERSC, ANL, and BNL.

Essential scientific goals of the participating experiments can only be met if the problems outlined in this proposal are solved. Many of the fundamental scientific issues involve large data sets that must be examined to search for rare occurrences. Such searches require the capabilities developed by this GCA.

2. Context and Background

The experience from current and past HENP experiments (CDF[3], D0[4]) show that scanning a TB or more of processed data on tape and selecting gigabyte subsets for detailed analysis is extremely cumbersome, sometimes taking days, or weeks, to answer "What if ?" questions. Past experiments have shown that appropriate trigger design (hardware selection of events for permanent storage) can match experimental goals to processing capacity. However, for many of the fundamental scientific issues, even after the data reductions provided by the trigger selection, the remaining data samples are quite large. This is necessary to ensure high signal efficiency. Rapid analysis of the data provides feedback to running experiments for "tuning" of trigger system parameters for optimal overall experiment performance. In this proposal we refer to data rates, sizes and event sample after appropriate trigger selections.

Previous and existing R&D projects (PASS[5] , RD45 at CERN[6] , CAP at FNAL[7]) lead to the conclusion that a software paradigm of dealing with physics data as objects (in C++) is much more effective than the common approach of unstructured data in FORTRAN. For this reason

the existing code base of the widely used CERN libraries are being frozen or even discontinued and all new software development for the LHC era at CERN will use C++.

This project will seek a strong coupling with computing activities at CERN regarding developments for the LHC era experiments both for the efficiency of leveraging development effort and knowledge as well as ensuring compatibility across the HENP community for which CERN has played a leading role.

Each of the current experiments participating in this project have facilities being planned at the respective accelerator labs to carry out the Event Reconstruction stage on the detector data (RHIC Computing at BNL, SLAC Computing Services, CEBAF central computing) and a data storage system. The amount of capacity on-site for physics analysis varies by laboratory. In every case the resources for simulations, and analysis of the simulated data, which are comparable to those for real data (except in data volume) are planned to come from off-site locations, either regional centers or desktop workstations.

2.1 Description of experiments

2.1.1 Solenoidal Tracker At RHIC (STAR)

The STAR experiment[8] at RHIC will seek to find a new deconfined phase of matter called the Quark Gluon Plasma (QGP)[9] . It may have existed approximately 10 μsec after the creation of the Universe and may exist in the cores of dense stars[10]. The goal at RHIC is to produce the QGP in ultrarelativistic (near light speed) collisions of heavy nuclei. In such collisions the nucleons in the nuclei will be heated and compressed to such high energies that they "melt", liberating their constituent partons (quarks and gluons) to interact freely within the plasma that they form together. STAR will measure hadronic signatures of the Quark Gluon Plasma, complementary to the PHENIX detector which concentrates on leptonic measurements. The uniqueness of STAR will be its capability of measuring the several thousands of charged hadrons that are emitted in a single event from near head-on collisions of nuclei. From these measurements STAR will then categorize each event based upon the values of the measured observables in order to identify special events, which exhibit the predicted characteristics of the formation of a QGP.

The collaboration consists of 35 institutions (23 of them US). A detailed description of the experimental setup, event reconstruction and physics analysis is presented in the Conceptual Design Report[11]. The STAR detector is under construction and is scheduled to start data taking in 1999. The experiment will take 10^7 events/year. It will generate a petabyte of data during the first years of data taking.

An especially challenging aspect of analysis is to identify subsets of events that contain evidence of dynamic fluctuations characteristic of the new physics and to distinguish them from the tails of the statistical fluctuations. A typical physics data set for STAR will be of the order of several percent of the yearly data production and a fully reconstructed data set will not be smaller than the raw data. The physics of STAR requires that several tens of such subsets are analyzed simultaneously. This makes development of efficient data scan methods, at various levels of reconstruction, absolutely necessary.

A large volume of the Monte Carlo data (of the same order of magnitude as the real data) is needed to develop reconstruction algorithms and investigate various statistical methods of data analysis. By the end of this year there will be ~10^5 fully simulated events available, which could be immediately processed using the hardware and software system proposed in this GCA.

2.1.2 Pioneering High Energy Nuclear Interaction eXperiment (PHENIX)

The PHENIX experiment[12] plans to make systematic and simultaneous measurements of several signatures that characterize the Quark-Hadron and chiral symmetry restoration phase transitions to examine if any or all of them show abrupt changes with energy density. Measurements will be made at several collision energies for proton-nucleus and nucleus-nucleus collisions. The experiment will in addition study nuclear collision dynamics, medium modifications of basic QCD processes, and the thermodynamic features of the collision volume. PHENIX is capable of measuring hadronic variables and is unique in its ability to study penetrating probes such as photons and lepton pairs. Since several of the proposed measurements in PHENIX involve rare processes, the experiment is designed to be capable of taking data at the highest luminosity expected at RHIC.

For PHENIX, not only is the shear amount of data a challenge, but also its complexity. Estimates are that as many as 1000 different objects will be stored per event. These objects will describe aspects of the measured data and the conditions under which they were obtained. Data analysis will require complicated cross correlations between many of the these objects . Some of these objects will be accessed repeatedly whereas others will only seldom be used. This implies that we need a sophisticated object hierarchy and methods for mapping this hierarchy on the data storage hardware in order to get the fastest possible access.

Also, PHENIX's need to search for rare events, and the need to analyze very large samples of data to extract di-lepton signals will require advanced data mining, data storage and manipulation techniques.

2.1.3 CEBAF Large Acceptance Spectrometer (CLAS)

Construction of the CEBAF Large Acceptance Spectrometer (CLAS)[13] and a tagged photon facility at the Thomas Jefferson National Accelerator Facility (TJNAF)[14] is scheduled for completion in October of 1996, with experiments expected to begin about six months later. The versatility of the CLAS and the wide range of beams available in Hall B will help exploit the capabilities of the electromagnetic interaction to explore the internal structure of hadrons, non-perturbative QCD, and the transition to the perturbative QCD regime.

Many of the studies will concentrate on the structure of the nucleon and its higher resonances. While these resonances have long been known, progress on their understanding has been slow. In fact the full spectroscopy and decay scheme of these resonances is still largely unknown because of their large overlap and the difficulty of untangling them. The CLAS studies will significantly increase the volume of electro-magnetic excitation data and complement the existing hadronic data to resolve many open questions. Not only will this data set unveil the complementary electro-magnetic couplings, but it will also provide a very high statistics, high quality, data set which can be used to independently identify and characterize the resonances by phase-shift analysis. Many additional physics topics will be investigated, such as strangeness production, electromagnetic sum rules of the nucleon, nucleon correlations and small components in nuclear systems, color transparency, and many aspects of meson production. These problems are uniformly rendered challenging by the fact that perturbative approximations of QCD are invalid in this energy range. On the other hand, this also adds to their interest because they may contribute to the understanding of the relevant degrees of freedom and techniques which can be applied in this non-perturbative regime which is fundamental to the understanding of such basic questions as the mass, size, and interactions of all baryons and mesons.

CLAS experimenters are expecting a sustained data rate of 10 MB/sec with an event size of 10 KB which will produce 150 TB of data per year for an operational period of one half year. As with the collider detectors, it is likely to operate in this mode for ten years. The TJNAF Computing Center facilities will be used by collaborators to perform the event reconstruction of the raw data. After removing noise data and adding calibration and reconstruction information, it is expected that the event size will be reduced by a factor of between 2 and 10. Hence, it is expected that reconstructed data will be generated at a rate of 100-500 GB per day. Approximately 70% of these events will consist of single electron or electron plus proton events which are of interest primarily for calibration analysis. The remaining 30% of the reconstructed events will be used for subsequent physics analysis - a rate of 30 to 150 GB per day, or up to 1 TB per week. It is this data that will be sent to NERSC for clustering and further analysis.

CLAS consists of many experiments running simultaneously. This is essentially similar to the RHIC and HEP experiments in which data for various physics topics are acquired at the same time. In the case of CLAS, however, each physics topic is formalized with a proposal to the TJNAF Program Advisory Committee. The amount of data to be used by the individual experiments vary. A few of the experiments will require physics analysis of the majority of the events stored at NERSC. The volume of data required for proper physics analysis will swamp any current computing facility.

Current plans call for a second phase of analysis to be conducted at TJNAF. It is expected that another order of magnitude reduction in DST size is possible during this second phase. The resulting data will be small enough that most experimenters will be able to carry out the subsequent analysis on their own computing facilities. Carrying out this phase requires clustering and analysis of the data as described in Section 3 of this proposal. There is no current software technology in the CLAS collaboration for clustering and analyzing the required amount of data. The research of this proposal will be invaluable to the CLAS data analysis.

In addition, this second phase of data analysis will consume extensive computing resources. The availability of the proposed infrastructure will greatly accelerate this second phase of data analysis and better utilize the DOE investment in TJNAF.

2.1.4 BABAR

The BaBar experiment is being built at the Stanford Linear Accelerator Center to study CP violation in the decay of B mesons. It will start taking data during early 1999, and is expected to record about 10^9 events a year. A running period of about 10 years is planned. A number of specific CP-violating reactions will be studied, with each typically occurring in a few thousand events per year. Separation of these rare signals from the much more common backgrounds will require careful reconstruction and analysis. High statistics studies of detector behavior are essential in getting the physics results out and are a difficult computational task. Given the required complexity of the event reconstruction and the need for robust and understandable software, the BaBar collaboration has adopted object-oriented design and programming techniques for their analysis software. This includes data storage in object-oriented database(s), and a fully objected-based analysis program in C++. This system is currently being designed, with initial implementations expected to be demonstrated in the Fall of 1996, and fully operational code by mid 1998. A key requirement is that reconstruction and analysis programs have efficient access to large amounts of data at different levels of detail. The solutions outlined in this GCA would be of immediate use in meeting this requirement.

2.1.5 ATLAS

The ATLAS Collaboration proposes to build and operate a general purpose proton-proton collision detector which is designed to exploit the full discovery potential of the Large Hadron Collider (LHC) at the CERN Laboratory in Switzerland. The collaboration consists of 140 institutions (25 US) world-wide. The principal scientific goal is to search for and study the mechanism behind electroweak symmetry breaking, generally believed to be the origin of the mass of elementary particles; thus of fundamental importance. The proposal has been approved, and the experiment will commence data collection and analysis in 2004. The experiment will generate one petaByte of scientific data per year during an expected operational period of 10 years. A very detailed description of the detector design, computing requirements and scientific analysis of the physics data is available in the "ATLAS Technical Proposal"[15]. The scientific challenges include the search for very rare "new phenomena" physics signals contained in very large backgrounds. In addition, the examination of some physics processes will require the study and manipulation of very large data sets. Both new physics analysis and computing techniques will need to be developed to handle these extreme conditions. During the design, construction, and commissioning phase of the ATLAS project a particularly challenging scientific activity will be the detailed Monte Carlo based simulations of the ATLAS detector to study the performance of the detector. In addition, the Monte Carlo simulation will be used to design, construct and evaluate computer algorithms and systems to perform the physics and statistical analysis of data produced by the detector for hundreds of potential new physics signatures. The computing challenges for the ATLAS experiment fall into the following categories: The filtering of extremely large raw data sets (1000 TB/year) to produce physics data sets which may range from 0.1 to 100 TB depending on the particular physics being studied. Typically from 50 to 100 overlapping physics data sets will be made available to the experimenters from the raw data sets. The manipulation, filtering and presentation of data sets of the scale 10 - 100 TB is particularly difficult from a storage and I/O access perspective. New techniques which allow the intelligent querying of the data, but minimize the motion of the data in a storage hierarchy will be required.

The access to the scientific data of large number of scientists (500 simultaneous users) distributed over the entire world is particularly challenging. New techniques based on distributed programming and object-oriented data bases, using transparent data access, caching, migration and replication are necessary. They will be built on a cooperating collection of regional computing and data storage centers using high-speed wide-area network connectivity.

The ATLAS simulation group is planning to prepare a full detector simulation with 10^7 jets. If each event is taken as 1 MB, this corresponds to the generation and storage of 10 Tbytes, but may be reduced somewhat by event selection prior to storage. This task will be shared by five or more sites with sufficient computing resources. ATLAS-US has proposed to do a share (1-2 TB) on the Parallel Distributed Supercomputer Facility (PDSF) at LBNL-NERSC. Thus a project which would provide storage and access (for analysis) of this size sample would be immediately used by the ATLAS-US collaborators.

2.1.6 Summary of major parameters of experiments.

The following table summarizes some of the important parameters of the experiments outlined above.

Table 1.

Collaboration	# members /institutions	Date of first data	# events/year	total data volume/year (TB)
STAR	350/35	1999	10^7-10^8	300
PHENIX	350/35	1999	10^9	600
BABAR	300/30	1999	10^9	80
CLAS	200/40	1997	10^{10}	300
ATLAS	1200/140	2004	10^9	2000

2.2 Computing Model

2.2.1 Processing model

Each of these experiments (STAR, PHENIX, BABAR, CLAS, ATLAS) has effectively the same overall data processing model. Even though there are variations at the detail level for each experiment the essential features can be described in the context of three stages of processing:

- Event Reconstruction. This is the primary reconstruction phase and converts the raw data into physical quantities such as hits, tracks, vertices, clusters of energy in the calorimeter etc. It might optionally classify events according to different physics signatures. This is usually performed once on each event but may be done more than once in the case of corrections or development of improved analysis techniques.
- Data Summary Tape (DST) Analysis. This stage is where events for different physics processes are separated into event samples, and where any physics-specific reconstruction will be performed (e.g. constrained vertices etc.). This stage is usually carried out separately a few times for each event. The number of times is highly dependent upon the physics topics of each experiment.
- Physics Analysis. This comprises multiple scans over an event sample, applying cuts based on an examination of the output statistics (in the form of histograms or similar visualization techniques), and iterating towards a final physics event sample. The resulting event samples are used for detailed analysis on particular physics processes. This stage is carried out many more times for each event than the DST Analysis stage.

In addition to the real data (from the detector) which is processed in these stages there is a comparable amount of Monte Carlo simulated data which is processed as well. This simulation data is used both to characterize the response of the detector (input at Event Reconstruction) to determine and characterize all types of background , and for comparison to physics models (input at Physics Analysis).

Figure 2.1 shows a schematic view of the essential features of the computing model. The form of this model is determined by physical and cost limitations (serial stream from detector, tape based storage of large data sets) and by the nature of physics analysis algorithms that require branches in processing of events depending upon the particular physics topic being addressed. The storage and access modes listed (store by feature, selection by feature, random access) are methods that will be developed by this GCA and form the primary motivation for it.

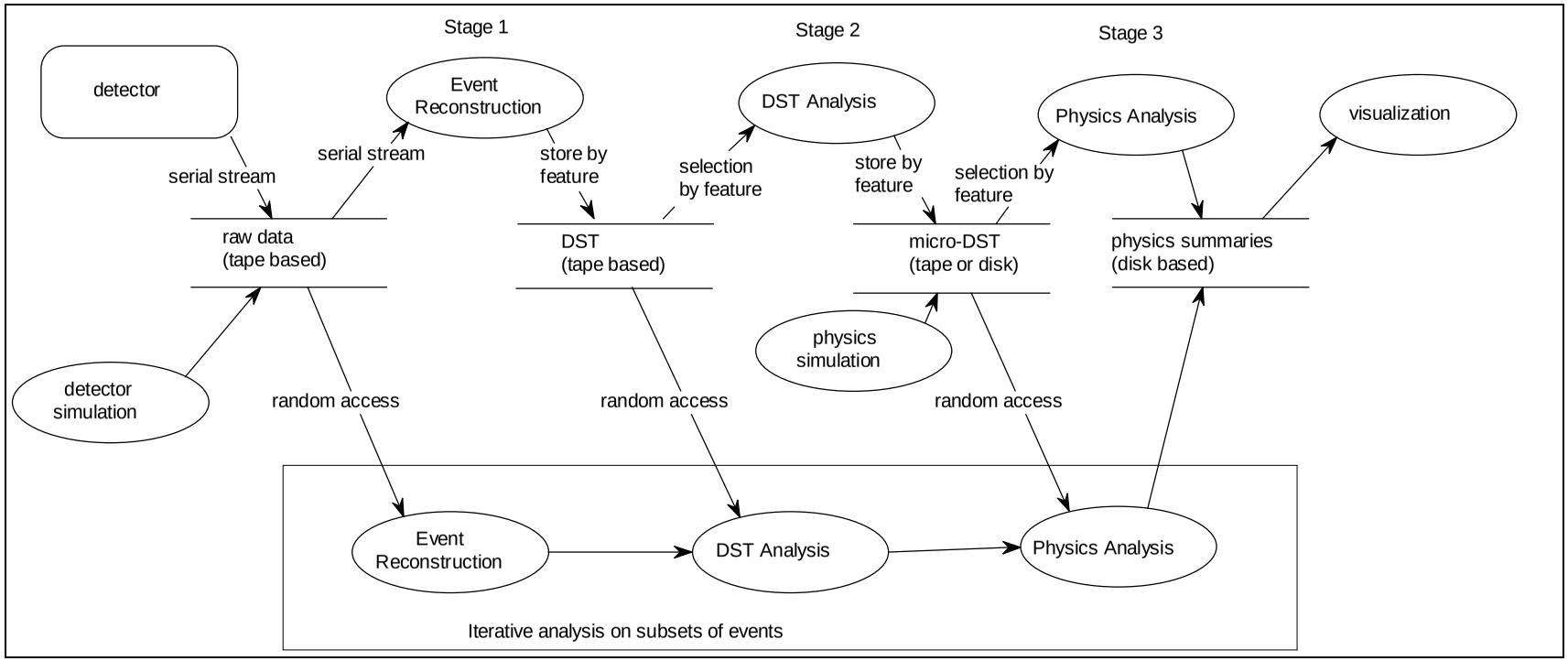


Figure 2.1. Simplified data flow view of computing model.
Event data is produced originally at the detector (raw data) or from a simulation of the detector response. These events pass through a first stage of processing (called event reconstruction) and the additional information generated is stored as DST data (historically called a data summary tape). A selected set of these events are processed further to generate derived quantities useful for physics analysis (like particle momenta), which are stored as micro-DST data. These micro-DST data sets typically form an enriched sample of events suited to a particular physics topic. The micro-DST event data is used in a final stage of processing (physics analysis) and summaries are generated which can be displayed with visualization tools.

During the physics analysis stage of processing a scientist will typically need to go back and review data and reprocess events from the earlier stages in order to verify or improve upon the algorithms used previously. This is indicated as the box labeled Iterative analysis on subsets of events.

2.2.2 DST access time to generate micro-DST dataset

The table below provides an illustrative example for the process of selecting from the DST event data in order to produce the micro-DST datasets. It shows the time in days and the effective read bandwidth for 100 people to access 50 TB of DST level data. The size of this example corresponds to that from a single experiment. Ideally, every individual can scan the DST dataset independently, however it is reasonable for a small group of people working on the same or very similar physics topics to share the produced micro-DST dataset. Since physicists typically need to scan the DST dataset many times in the course of a particular physics analysis study this time scale should be fairly short but a few days is acceptable. Scenario 5 is considered to be an effective performance goal and would enable analysis to proceed without undue burden imposed by ineffective computing resources.

Table II lists various scenarios (column 1) according to how many people share data from one selection (a micro-DST) (column 2), how many tape drives are used in parallel (column 7), the fraction of the full 50 TB DST dataset which is selected (column 5), and the efficiency with which data can be read from the tapes in sampling mode (column 6).

While the user view of how the data is organized should be decoupled from the details of the mass storage system configurations, scenarios 7 & 8 illustrate the need to organize the data in the mass store (on tape) for efficient access. If the time is determined largely by the tape mount and head seek time rather than the data reading time then the efficiency scales inversely to the sample fraction (proportional to number of tapes touched) rather than the data volume.

Table II.

Scenario	# people	# people/ micro-DST	DST (TB)	% of data sample	sample efficiency	# tape drives	MB/sec	days
1	100	1	50	1	1	1	10	5787.04
2	100	1	50	1	1	10	100	578.70
3	100	10	50	1	1	10	100	57.87
4	100	3	50	0.1	0.5	10	50	38.58
5	**100**	**3**	**50**	**0.02**	**0.5**	**10**	**50**	**7.72**
6	100	3	50	0.001	0.5	10	50	0.39
7	100	3	50	0.02	0.05	10	5	77.16
8	100	3	50	0.001	0.005	10	0.5	38.58

2.2.3 CPU required for DST Analysis

The amount of cpu time required in the DST Analysis phase varies greatly depending upon the physics topic, the experiment, and the level of analysis completed in the Event Reconstruction pass. However, based upon benchmark studies for STAR [16] and performance monitoring data from the PIAF facility[17] at CERN we get a range of values (scaled to 50 MB/sec DST read bandwidth) from 0.7 GFLOPS for reading data, to 1.3 GFLOPS for minimal analysis to 30 GFLOPS for significant analysis. This yields an estimated cpu need of 1 to 30 GFLOPS for scenario 5 above.

2.2.4 Physics Analysis of micro-DST datasets

Reading micro-DST datasets and carrying out physics algorithm computations will be performed independently by individual physicists. Some datasets (< 100 GB) and cpu requirements (< 1 GFLOP) are small enough that the analysis can be carried out on individual workstations. There are, however, many physics studies (particularly for the RHIC program and later on for experiments at the ATLAS scale) that require micro-DST's in the TB range and significant amounts of cpu cycles which are the subject of this proposal.

A single physics computation pass on a micro-DST level should be the place where scientists are trying to answer the individual "What if ?" questions. "What if I select on these parameters?", "What if I try this set of secondary track fitting values, etc.?". In order to have the time scale for these studies to be set by the scientists creativity rather than computing resources limitations a single pass on a micro-DST should be completed in a few hours (or less).

The range of cpu required (per MB of micro-DST data) varies significantly, even more than for the DST analysis mentioned above. The lower limit is set by the data read value (about 1 GFLOP per 50 MB/sec above). The upper limit depends strongly on the type of the analysis and may reach, in the extreme cases of cpu-intensive tasks (e.g. HBT analysis at STAR), the range of 30 GFLOP/MB/sec.

In order to process a 1 TB micro-DST in 3 hours (93 MB/sec) cpu in the range of 2 GFLOPS to 2800 GFLOPS is needed. A value of 2.8 TeraFlops (per scientist) is beyond the realm of any reasonable expectation in the next few years so it is clear that analyses of this scale must be approached by other methods and lie beyond the scope of this proposal.

In conclusion, we estimate that with a few people proceeding with the cpu-intensive analysis and together with around 100 people running less cpu intensive calculations, the scale of cpu required for effective physics analysis is in the few 100 GFLOP range.

2.2.5 Geographical Model

Each of the experiments participating in this project have a similar geographical model for computing resources, illustrated schematically in Fig. 2.2 Event Reconstruction and storage of raw data is located at the host accelerator laboratory. The major simulation effort is located at one or more regional centers. Software development for all processing stages along with some simulations and analysis of small datasets is carried out on desktop workstations located at most of the collaborating institutions.

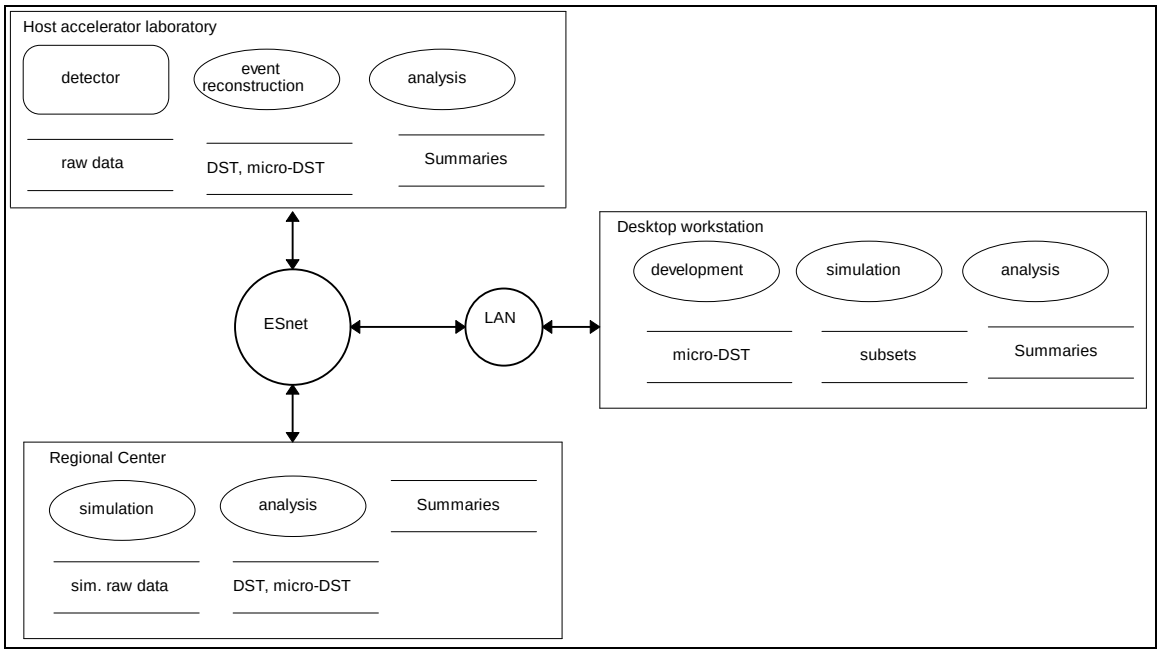


Figure 2.2. Geographical Model
The locations of computing activities are identified by the boxes. The dominant computing process types and data stores are indicated at each type of site. The host accelerator laboratory, the regional center(s) (like NERSC and ANL-CCST) are connected via ESnet. Individual desktop computers are connected to a local area network (LAN) which is itself connected to ESnet.

2.2.6 Details of experiments

The following table summarizes some of the important data reconstruction and analysis parameters of the participating experiments.

Item	**STAR**	**PHENIX**	**BABAR**	**CLAS**	**ATLAS**
raw event rate (Hz)	1	67	100	1200	100
raw event size (MB)	16	0.3	0.025	0.01	1.3
<raw rate> (MB/sec)	16	20	2.5	10	130
# event/year	10^7	10^9	10^9	10^{10}	10^9
raw size/year (TB)	230	288	25	200	1000
GFLOPS-sec/evt. Evt. Recon.	33	0.4	.04	0.004	2.5
DST event size (MB)	2	0.1	.01	0.02	.13

3. Approach

The problems described above, access to and analysis of massive data sets, are seen within the HENP community as the most demanding computing issue impeding the advancement of the scientific goals in nuclear physics and high-energy physics. Our approach to solving this problem is to bring together a multi-institutional multi-disciplinary team in which the members have a history of expertise in the technical areas to be addressed. In addition, the strongest assurance that can be made regarding the success of this effort is the vital coupling with the HENP experimental groups involved who view the success of this project as necessary for the full achievement of their goals.

Most of the technical areas to be addressed fall into the category of software infrastructure. A brief description of some of the details in these areas is given below in sections 3.1-3.5. In addition to the software infrastructure, which applies equally well to each experiment, there are efforts related to each particular experiment, which is outlined in section 3.6.

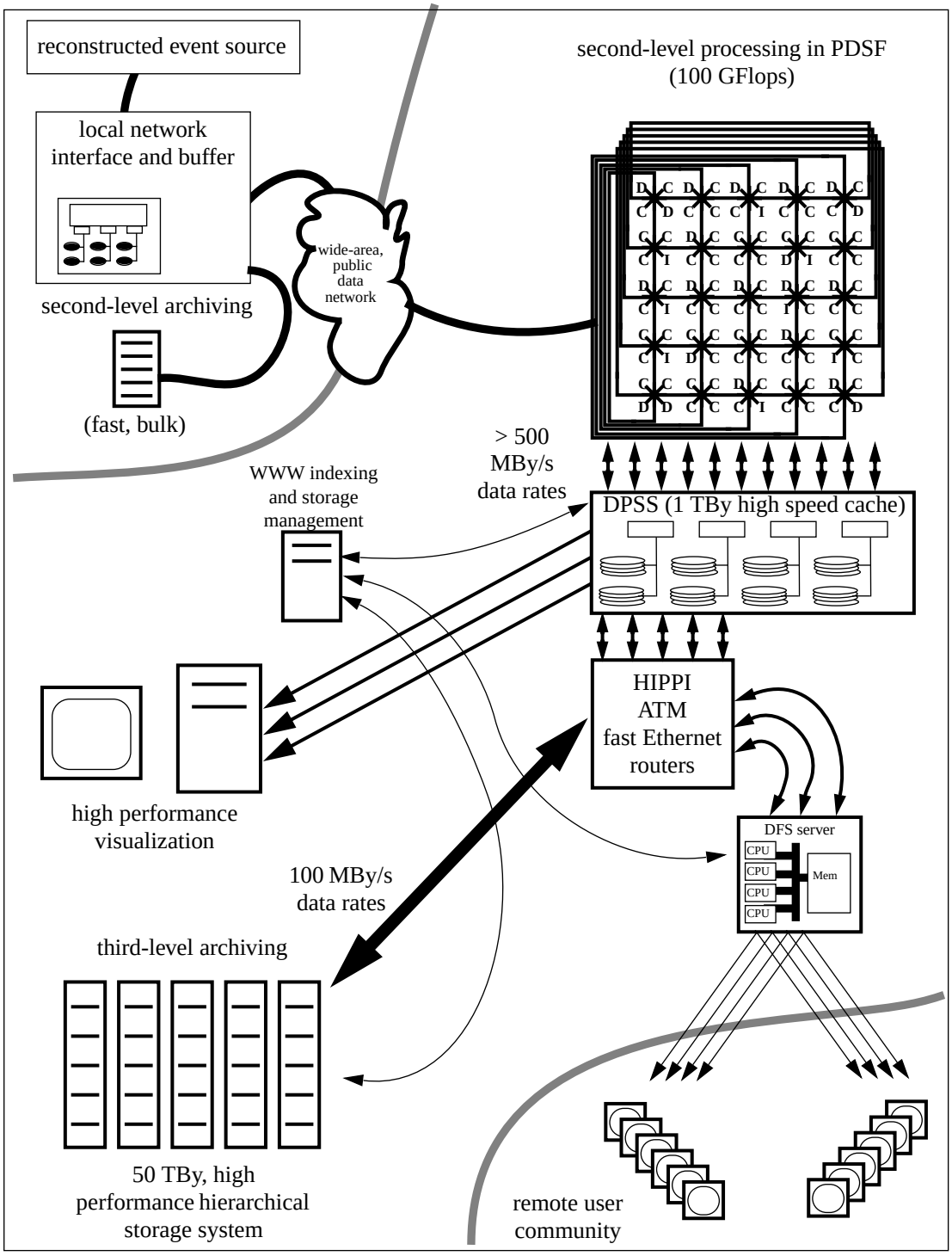
As indicated in the schedule (section 5.3) the first stage at the start of this project is to develop the detailed plan of work to be done and delineation of the interfaces which will be required for this collaborative effort to proceed efficiently. The technical areas we have identified as components of this project are:

- file storage system - hierarchical storage system with capability of application level control of storage organization and order of access.
- wide area data access - data access from desktop systems around the US (via Distributed File Server, DFS[18])
- persistent object system - optimal C++ classes for experimental physics objects (ODMG-93) [ref]defining interface to permanent storage methods.
- event-parallel physics computation with parallel data access- high performance computation framework for executing physics analysis algorithms
- user interface & visualization - user interface to visualization (data browser, data selection mechanisms, histogramming & fitting)
- experiment interfaces - interfaces to experiment specific storage models.

In each of these areas the team members have previous related projects with existing software which can either be contributed directly or serve as a base for modification and integration. This project is, in large part, an integration effort which should result in a production system that is used at NERSC, ANL and BNL.

A schematic view of the hardware at NERSC which will be used is shown in Figure 3.1.0. The major hardware components which need to be added to the existing NERSC infrastructure to meet the goals of this proposal are:

- cpu farm (likely SUN SMP) (100 GFLOPS peak)
- IBM 3590 tape robotic system with controllers and HIPPI interface and Unitree host (50 TB tape, 10 TB disk)
- DFS server (1 TB disk)
- Suitable network interconnections for these systems as shown schematically in figure 3.1.0.



PDSF and Storage System, Physical Architecture

Figure 3.1.0

Reconstructed events (upper left hand corner of fig. 3.1.0) are passed into the PDSF and subsequently stored in the mass storage system (MSS) according to a data indexing protocol to be developed (see section 3.1). In addition, the PDSF farm will be used to generate the necessary simulated DST's. Analyses on archived real and simulated DST's will be carried out on the PDSF, and with local computing capacity by copying the data from MSS to DFS. The DFS system thus serves to handle data traffic between the MSS and individual users. This traffic is supervised by the WWW indexing and storage management server. The performance visualization module allows individual events taken directly from the Distributed Parallel Storage System (DPSS) to be displayed and inspected.

3.1 Optimization of Storage Layout and Access

The purpose of this component of the proposal is to develop the necessary technology for the efficient access of the mass storage system (MSS). Such technology is essential given the massive quantities of data that have to be stored and selectively accessed. Because of the large amounts of event data collected from these HENP experiments, it is impractical to store them on disk systems; they must be stored on tapes. The access from these tape systems form a major bottleneck because of the inherent sequential nature of the devices (requiring long seek times to reach the desired events), and the time to mount and dismount tapes. The problem that needs to be addressed is how to retrieve desired subsets of the events for analysis from the tape storage system in the most efficient manner. The approach is to organize the event data on tape storage in a way most appropriate to the probable access patterns of the data rather than the order in which they were generated.

We concentrate here on the storage and access needs of the processing model described in the introduction, and the processing steps shown in Figure 2.1. To recapture, after the raw events data is generated and stored on tapes, the process of analyzing and selecting event data for further processing has three major steps:

1) The event reconstruction step (usually called "pass1"), which analyzes the raw data, identifies the particles of each event and their properties, and generates summary data in a form referred to as DST (Data Summary Tape).

2) The DST analysis step (usually called "pass2"), which selects subsets of the DSTs according to desired properties, and generates smaller collections of events (often called "micro-DSTs").

3) The physics analysis step, where physicists search a given micro-DST according to features of the particles in each event, and produce summaries. These summaries are usually visualized as part of the analysis process.

3.1.1 Storage layout and indexing

The main idea of our approach is to order the events on tapes according to their properties, so as to minimize the number of tapes that need to be mounted and the number of tape seeks made when subsets of the events are needed. As we discuss below, this is most appropriate for steps 2 and 3, but not step 1. In step 1 all the data need to be processed, and thus parallel access and parallel processing can efficiently be used. Since all the events need to be accessed there is nothing to be gained from reordering the raw data.

In step 2, only a small fraction of the data is needed at any one time to generate the micro-DSTs. Extracting the desired fraction may take many hours, even days, if the data is not properly organized and indexed. Our strategy is to apply "data layout" algorithms that will optimize the placement of the events on tape in order to minimize access time to the subset of events. This

will be achieved by using multi-dimensional indexing methods on the properties characterizing the events, and placing the events in storage according to their distribution in this multi-dimensional space. This is described in Section 3.1.6 below.

In step 3, the problem is to select events from micro-DST according to selected properties of their particles and global event observables. Often, the micro-DSTs are sufficiently small (10-100 GB) that they can reside on the user's disks. However, some micro-DSTs can be as large as 1TB, in which case it is more reasonable to store them on a shared mass storage facility, such as NERSC. In such cases, one can design the system for sharing the micro-DSTs by several physicists, each one interested in a different subset of the properties of these events. For micro-DSTs stored on disks, one can take advantage of parallel access. Thus, the problem is one of distributing the events on parallel disks to maximize throughput for the most likely queries. In addition, it is necessary to build an efficient index to the properties of particles. Although each event can have in some experiments thousands of particles, and each particle can have several properties per event, the number of properties that each physicist is interested typically in a dozen. Thus, a multi-dimensional indexing methodology for about 10-20 properties can be employed here (see section 3.1.4). However, in this case we will need to support multiple indexes on the same micro-DST, to accommodate the needs of various analyses performed by several physicists accessing events in the same micro-DSTs.

3.1.2 Interfacing to the mass storage system

In order to achieve the benefits expected from the organization of event data on the mass storage system, it is necessary to have control over the placement of the data, as well as ways to read the data in the desired order according to the application's needs. At a minimum, the following two types of control are necessary:

1) When placing files, it is necessary to dictate to the mass storage system which files should go on which tape and in what order.

2) When reading files, it is necessary to ask the system to read a set of files (which are possibly on multiple tapes), and have the system read them in a way that minimizes mounting and dismounting of tapes, as well as read head seeks. That means that the mass storage system will sort out the request, and schedule all files that reside on the same tape to be read sequentially.

Unfortunately, the software that drives current mass storage systems (such as Unitree) usually does not give this level of control to the application software that interacts with it. The protocols provided assume a single file at a time interface, where the system has full control on where to place files according to its own optimization strategy. Our strategy is to work with the NERSC staff to accommodate these needs using the current NSL-Unitree system. As is described in section 3.1.6, we plan to take advantage of some facilities in the NSL-Unitree system to get around these problems in the short term. In addition, changes to the NSL-Unitree software will be made by the NERSC staff as necessary. In the long term, we expect that such control over the placement and reading of files will be available when a new mass storage software system, High Performance Storage System (HPSS), will be installed at NERSC.

3.1.3 Efficient retrieval of micro-DSTs (step 2)

As mentioned previously, one of the goals of this proposal is to facilitate query-by-feature from very large event databases. Such features are extracted in the event reconstruction step (pass1). Typically 10-20 such features per event are extracted, such as the number of each particle type in the event (e.g. number of pions, kaons, protons, electrons), the transverse energy, the "temperature" (shape of the spectrum of the momentum), etc. Placing these events in the multi-

dimensional feature domain, generates correlated distributions of the events. The correlated distributions depend on the type of experiments and the type of features of interest. For example, in particle physics experiments, if we view the 2-dimensional space of p (momentum) vs. dE/dx (energy loss) of particles, the distribution of the events in this space could be imagined to look like "cloud" clusters as shown in Figure 3.1.1. Another example of a different sort can be found in nuclear heavy ion experiments. If we view the 2-dimensional space of E_T (transverse energy) vs. the N (the number of pions), the distribution tends to be along the diagonal as shown schematically in Figure 3.1.2. Note that the concentration of events in this case is in the middle of the band, because each property domain tends to have a bell shaped distribution.

In general, the multi-dimensional space will be sparsely populated where events with similar features form clusters. Thus, the storage of the events according to these clusters in the multi-dimensional feature space will maximize the likelihood that queries based on features will end up retrieving events that are stored close to each other on tapes.

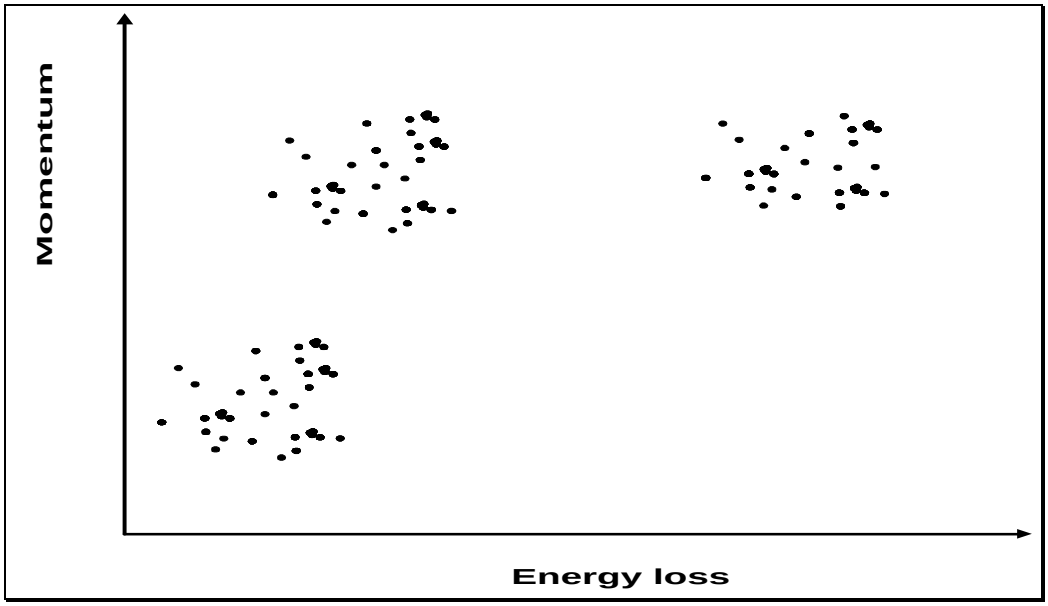


Figure 3.1.1: An example of a distribution of events in feature space for particle physics experiments

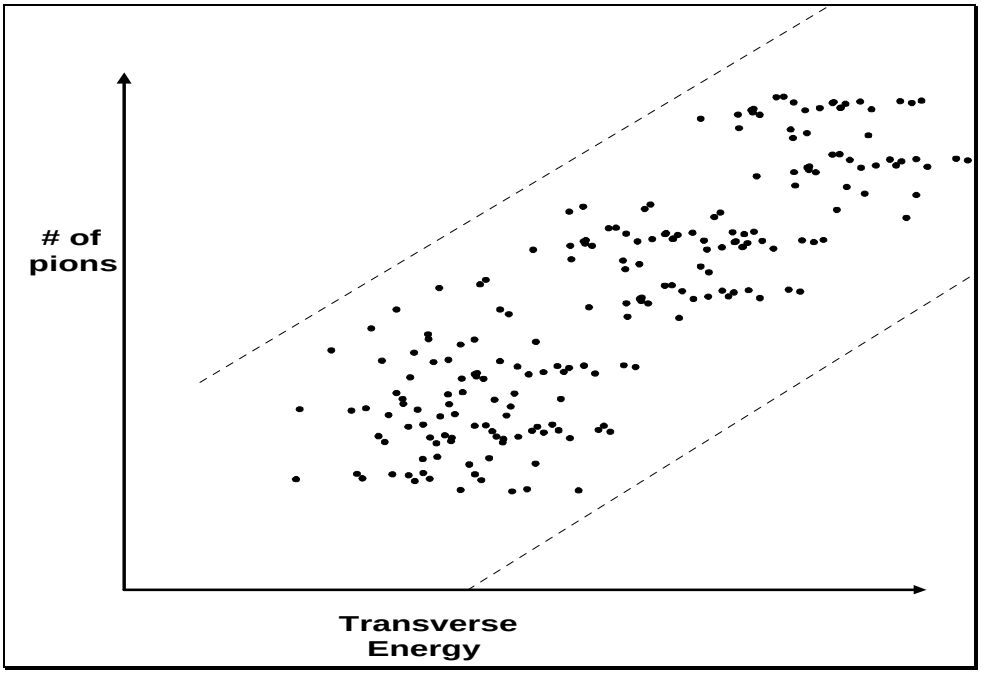


Figure 3.1.2: An example of a distribution of events in feature space for nuclear heavy ion experiments

3.1.4 Indexing structures

The second part of our strategy is to use an index for locating the desired events in response to feature-based queries. We would like to focus on requests for events that have certain features or a range of features. For this purpose, we will use a multi-dimensional (MD) index on the feature space. The problem here is that the feature space is likely to be very sparse, and the number of dimensions is quite large (10-20 features on a space of up to 10^9 events). There are some indexing methods known for sparse multi-dimensional datasets, such as Grid Files, multi-level B-trees, K-D trees. However, these methods can be quite inefficient when the number of dimensions is larger than 4-5. It is an open research problem to partition a high multidimensional space into cells such that each cell contains elements that are likely to be access together. The goal is to devise an indexing method such that the percentage of relevant elements in each cell retrieved is high.

One possible approach is to use "binning" and "partitioning" methods for the indexes. "Binning" refers to the process of breaking a continuous domain into a number of ranges. For example, "energy" can be binned into 4 categories for a particular experiment: <10, 10-20, 20-30, >40. The flexibility to choose the smallest number of bins for an application can simplify the index. "Partitioning" of an index is the pre-selection of properties into several groups. The properties that are likely to be accessed together (i.e. conditions applied to them in queries) are put into the same group. In this way one can support a 20-dimensional index by several partitions of lower dimensionality. Feedback from experimenters and characteristics of acquired data will guide the partitioning scheme and the choice of the feature variables for individual experiments. This is shown schematically in Figure 3.1.3, where the original multi-dimensional feature space (10-20 features), is partitioned into groups of features which form several lower dimensionality feature spaces. We will explore and simulate various multi-dimensional indexing methods, and adapt or devise an indexing method most appropriate to the features extracted from events. In addition, a

monitoring package will be provided to allow measurements and analysis of system performance. Such measurements can be compared with model estimates to provide feedback on the selection criteria. This index will then be implemented as part of the access tools to the MSS, to determine the location of the desired events.

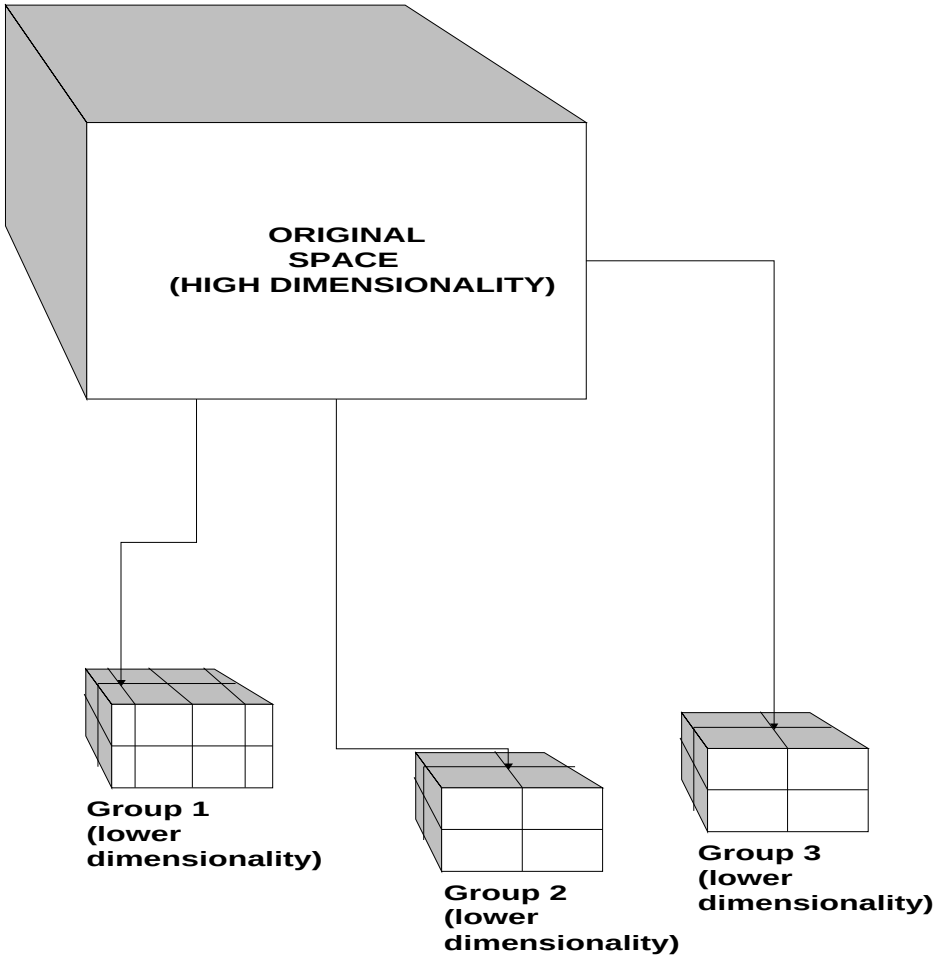


Figure 3.1.3. Partitioning of the index into several groups, and binning the values on each dimension

Multi-dimensional (MD) indexing methods essentially partition the MD space into tiles. Depending on the MD indexing methods, the tiles may have various shapes and sizes. For example, one can use adaptive partitioning methods, such as the "grid file" to partition the 2-dimensional example in Figure 3.1.1, with lines crossing horizontally and vertically. Partition lines cut through dense regions so that the average number of points (representing events) is roughly the same in each tile. This is shown schematically in Figure 3.1.4. The main idea of the data layout algorithm is to try and preserve the proximity of tiles in the MD domain when they

are stored on tertiary storage. Thus, events that are close together in the MD feature space will be stored in adjacent locations on tape. The index will serve as a quick and efficient way of locating each event using the tiles structure.

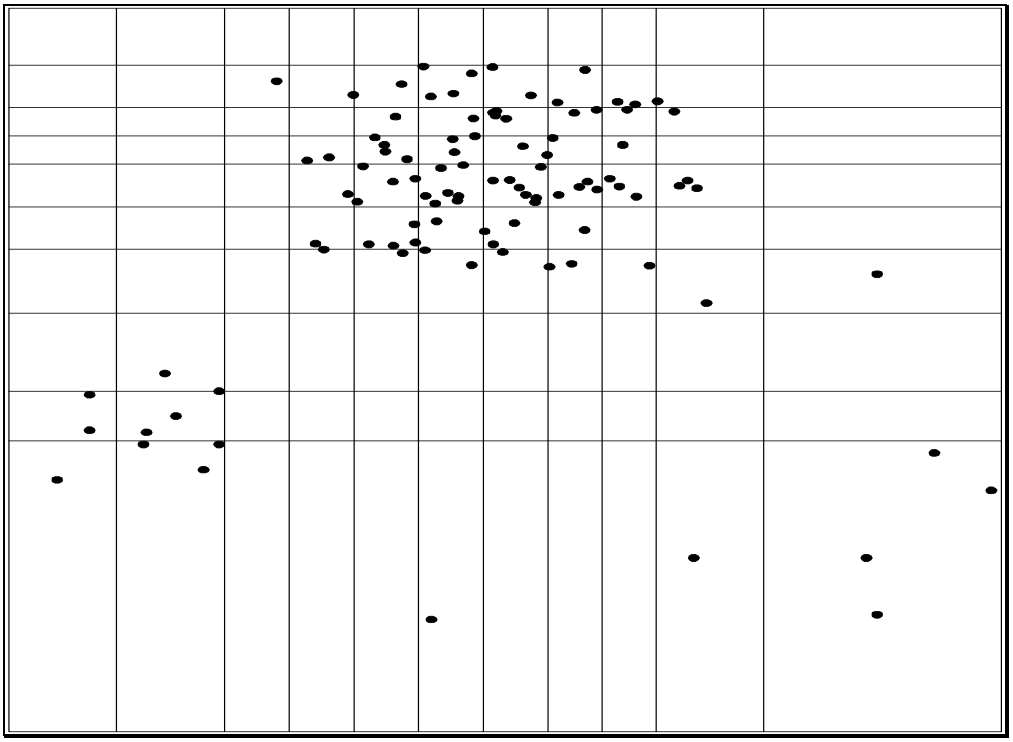


Figure 3.1.4. Adaptive partitioning, more tiles in dense regions.

The size of the tiles will also depend on the efficiency of the mass storage system in accessing small files. At one extreme one can put a single event in a tile, where each tile is stored as a file. The overhead of reading a large number of events (i.e. files in this case) will be high. At the other extreme, putting too many events in a single tile will cause reading extra events when a few events are needed from a single tile. The optimization problem for selecting the appropriate tile size will be addressed as well.

The type of indexing method that is most efficient depends on the distribution of the events in the multi-dimensional feature space. The goal is to minimize tiles that are mostly empty (for very sparse regions of the multi-dimensional space). Thus for the example of Figure 3.1.2 where the distribution is along the diagonal one can apply a transformation (in this case rotation and translation of coordinates axes) before a partitioning index is applied. This transformation eliminates large empty regions from the index space. This is shown schematically in Figure 3.1.5. The cost for using such transformation techniques is a small additional processing to performe the reverse transformation when the data is retrieved.

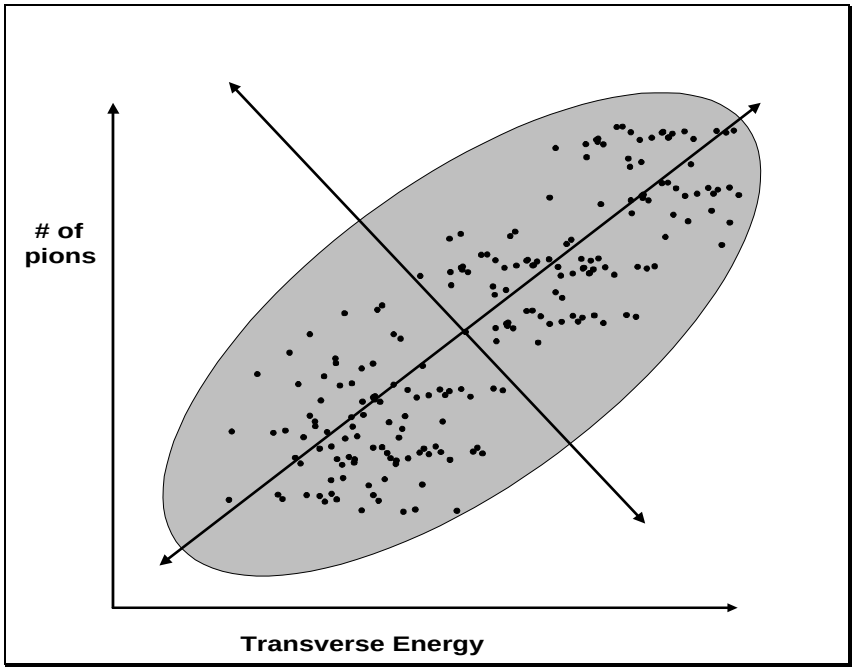


Figure 3.1.5. Transformation of coordinates to achieve better efficiency of an index.

3.1.5 Efficient layout of and access from micro-DSTs (step 3)

The problem is somewhat different in step 3, where the micro-DSTs may be small enough to reside on disks. While the same index methods can be used to partition the multi-dimensional property space, instead of storing adjacent tiles close to each other, we try to distribute them on disks for parallel access. Furthermore, the events contained in each tile should be distributed into as many parallel disks as possible. We have developed and used such techniques in the past for parallel access from disk systems in the context of the MAGIC project[19]. The method developed was especially suited for visualization as it guaranteed an upper bound on access time. We believe that such techniques will be useful in the context of this proposal.

Another aspect of supporting micro-DSTs is the need for multiple indexes on each micro-DST when they are shared. As mentioned above, the selection of events from micro-DSTs is based on properties of the particles and/or of the events as a whole.

Since each event may have thousands of particles (depending on the type of experiment) the space of particles is much larger than the space of properties of events as a whole. However, it is typical that each physicist may be interested in a small subset of these particle-level properties, in the order of 10-20. Thus, an index can be built for a given subset of the properties. Since multiple indexes may be built on the same micro-DST, there may be conflicting requirements on the organization of the micro-DST depending on which properties are selected for indexing. The solution to this problem results in an optimization that achieves a compromise between multiple ways of accessing the events.

Sharing micro-DSTs also brings an opportunity to share events that appear in multiple micro-DSTs. Rather than storing these events several times, one for each micro-DST they belong to, one may consider a global optimization scheme to minimize such duplication. These problems will be investigated and algorithms developed.

3.1.6 Details of the interface to the mass storage system

The analysis of the multi-dimensional features space of the events will be used to determine the layout of the events on the tertiary storage. There are two aspects of the interface to the mass storage system that need to be supported. On the one hand, it is necessary to instruct the mass storage system where to place the data. On the other hand, at the time that a data query is issued, it is necessary to issue access commands to the MSS to read multiple files that may reside on multiple tapes [See figure 3.1.0]. We discuss next how we plan to approach each need with the current National Storage Laboratory (NSL)-Unitree system available at NERSC.

There are no facilities in National Storage Laboratory (NSL)-Unitree to place files on a certain tape in a certain order. However, NSL-Unitree allows users to define a "family" of files, and to request that a file is placed in that family. In addition, it is possible for the system administrator to set aside tapes exclusively for an application (a job). Usually, NSL-Unitree places all files that are written to the mass storage system in disk cache, and then chooses when and where to migrate the files to tape. However, if the files belong to the same family, NSL-Unitree migrates the files according to their time stamp. If, in addition, tapes were set aside for the application, then only the application files will be written to these tapes. Thus, by using the combination of exclusive use of tapes and the "family" capability, one can control the placement of files in the desired order, as long as they are written out in that order. In addition, we can take advantage of the ability to inquire NSL-Unitree as to the locations of files, thus verifying after the fact that files were placed as requested. We will develop a module (called the "Data Layout" module) that can be called by any application to place files on tapes in a desired order. This module will also generate a "Data Layout Table" that will be used at the time the application needs to read files as will be described next.

As for reading a set of files in the most efficient manner, again there are no facilities in NSL-Unitree to do that directly. Thus, the software that reads files from the mass storage system on behalf of application program needs to know where files are stored (i.e. which tapes, and what location on tape). As mentioned above this information is generated by the "Data Layout" module and stored in the a "Data Layout Table". We plan to develop a module (called the "Data Retrieval" module) that will use this table to read files from tapes in the most efficient manner. Given a request for a set of files, this module will sort out the components of the request by tape, and by location within a tape. It will then issue the read request in order per tape. We plan to take advantage here of the way the NSL-Unitree system queues and processes requests. NSL-Unitree places requests in a queue, and determines which tape to mount according to its optimization and policy algorithms. One feature of these algorithms is that if a tape is mounted and there is another request on the queue to read a file from the same tape, it will attempt to service this request first in order to minimize tape dismounts. Thus, our plan is to schedule the read requests successively, so that they end up on the scheduling queue at the same time.

The above techniques of using the current facilities in NSL-Unitree to achieve the desired effect have not been checked and tested. We have coordinated this effort with the staff of NERSC, so that if we find any difficulties in achieving these goals, the NERSC staff will modify the NSL-Unitree software as needed. The testing of the NSL-Unitree capabilities (especially the "family" structure and whether the queue scheduling algorithm performs as expected), will be done early on in this project.

In a second phase of the project, scheduled to take place when a stable version of HPSS is installed at NERSC (expected at the summer of 1997), we will explore the ability of HPSS to perform the above functions. To the extent that this is possible, it will simplify the code of the "Data Layout" and the "Data Retrieval" modules.

In addition, it is expected that HPSS will permit the scheduling of mounting multiple tapes for the same job to achieve simultaneous reading of multiple tapes (i.e. parallel tape reads). We will test the effectiveness of this capability, and accordingly modify out data layout algorithms to take advantage of parallel tape reads.

3.1.7 Related Research

We expect to leverage the experience gained from a joint project, called OPTIMASS (for Optimization on Mass Storage Systems), which is a collaboration between LBNL, LLNL, and the University of Maryland. This project, which is in its third year resulted in the development and implementation of partitioning algorithms for a specific application: simulation data generated by climate models [20,21,22]. In that context, we have identified specific access patterns that we represented as classes of query types. We have shown in simulation results and in actual experiments that reorganizing the datasets into clusters can result in one or even two orders of magnitude of improved access time for the pre-specified query types. In general, the problem is one of finding the best compromise in how to store the data for conflicting access patterns. We have shown that although the general problem is NP-complete, it is possible to develop effective solutions using dynamic programming techniques. In addition, we have developed techniques for requesting these clusters in parallel, and extracting the desired subsets from them. Another important component of this work was the development of software to automatically reorganize a dataset from the specifications generated the optimization algorithm.

Another aspect of this project is the way application programs interact with the MSS. This aspect of the work was done in collaboration with the National Storage Laboratory (NSL) in LLNL. We have shown that in order to support efficient access to MSSs, it is necessary to support the storage and access of multiple "clusters" in a single request. We have run experiments with real datasets on the AMPEX robotic tape server that confirmed our simulated expectations.

Another project at LBNL relevant to the proposed work, is the development of an algorithm (and the software to implement it) for distributing 2-dimensional tiles for parallel access from multiple disk systems. This work was part of the MAGIC gigabit testbed project [23]. While existing algorithms cannot guarantee a lower bound performance for localized queries for adjacent tiles, this new algorithm guarantees a lower bound of less than 7% of optimal. This algorithm was later generalized to tiled data in higher dimensions as well. We expect to apply this algorithm to datasets that are small enough to fit on disk clusters, or to subsets of datasets that were downloaded from tertiary storage.

3.2 Wide area data access

An important aspect of collaborative research is the ability to share data across the range of institutions in a collaboration. Actually doing this across a wide geographical area (around the US) raises a number of difficult technical issues, like data synchronization and authentication (security). Fortunately, the computing industry has been addressing these issues for some time and viable commercial solutions to this problem are just becoming available. The Distributed File System (DFS), a higher level component of the Distributed Computing Environment (DCE), is just such a solution. DFS provides reliable data transport over wide area networks.

Access to data from a remote client is done efficiently by having a cache of file blocks locally on the client. As long as the client only needs data from within the blocks resident in the local cache this data is accessed with the speed of the locally attached disk on which the cache resides. If, on the other hand, the client needs data which is not resident in the cache then those blocks are transferred from the server at whatever speed the network will support. The synchronization of

the cache on the client with the actual data source on the server is maintained automatically by the DFS software.

The granularity with which the cache is synchronized with the server (file synchronized at the time a file is closed) can lead to inconsistencies for multi-user update access to files. However, most of the data for HENP experiments is WORM (Write Once Read Many) and the properties of the DFS caching are well suited to this type of access. Not all data is WORM, so care must be taken to minimize and properly account for the data which requires simultaneous multi-user updates. This area is probably best handled with network access to a commercial database system.

The issue of the proper handling of WORM vs. multi-user update data for HENP experiments will be investigated as part of this GCA and appropriate solution determined.

For wide area desktop access to an hierarchical storage management (HSM) system a possible long term solution would be to have DFS integrated with the HSM. It may be that such a solution will become available during the period of this GCA project in which case we would try to adopt it.

However, for the immediate future, the mechanism we propose using is to operate a DFS server as a separate system from the mass storage system (MSS) (initially Unitree at NERSC) and to supply the mechanism to trigger migration of files between DFS and MSS. This option is indicated in Figure 3.1.0

3.3 Persistent object system

There are often good reasons for designers of large scientific data stores to consider lightweight approaches to object data persistence. Prominent among these reasons are performance, scalability (data requirements may be in petabyte ranges), portability, adaptability to specific high-performance architectures, and price. Moreover, data access is often anticipated to be primarily from inside of user-written, numerically intensive programs. Access characteristics are usually close to Write Once, Read Many times (WORM), so elaborate locking and transaction mechanisms may be unnecessary. On the other hand, there is a well-founded fear of nonstandard, home-grown solutions, and concerns about the adequacy of data protection and integrity in non-database solutions. All of these factors apply to our target physics applications.

One attractive strategy in many cases is to use a lightweight persistence manager today to exploit special-purpose architectures or to otherwise meet performance demands, while leaving open the possibility of migrating data to true object databases as commercial products begin to provide the performance or scale or portability that applications require. Another is to support an architecture in which some data are maintained in true object databases, while other data reside in a persistent store, according to the application's requirements. One problem with both of these strategies has been that choosing to use object persistence managers has often meant writing code and class definitions entirely different than those expected by true object databases. With the emergence of object database standards, many of these differences are needless handicaps--it is a disservice to users of lightweight persistence managers to make coexistence with, or migration to, object databases unduly difficult.

3.3.1 ODMG details

The Object Database Management Group (ODMG) is an industry consortium of database vendors and others who have come together to agree upon aspects of a common specification for object databases. These efforts have resulted in an emerging standard (currently ODMG-93 Release 1.2[24]) whose components include: an object model; an Object Definition Language

(ODL); an Object Query Language (OQL); a C++ binding for ODL and OQL, and a C++ Object Manipulation Language; a Smalltalk binding for ODL and OQL, and a Smalltalk Object Manipulation Language. While ODMG-93 is an object database specification, a significant subset of it can be supported in a natural way by many lightweight object persistence managers.

ODMG's basic modeling primitives are objects, which have identity, and literals, which do not. The state of an object is described by its attributes, and by the relationships with other objects in which it participates. An object's behavior is specified by the operations it supports. Objects are instances of object types. A type has an interface specification and one or more implementation specifications; like many object models, the ODMG model supports inheritance-based type-subtype relationships.

In the ODMG model, objects are allocated and persist in storage domains known as Databases. Databases support text-based object naming as a means to access particular objects, and to begin navigation through object databases. Access to objects occurs within the scope of a Transaction. Transactions are the means by which ODMG guarantees atomicity, consistency, isolation, and durability. The model uses a conventional lock-based approach to concurrency, and supports pessimistic concurrency control as a default policy.

The ODMG's Object Definition Language (ODL) is intended to be compatible with the Object Management Group's Interface Definition Language (IDL). ODL extends IDL by defining parameterized types for collections, e.g., Set<T>, Bag<T>, List<T>, and Array<T>, for collections of objects of type T, by introducing keywords for persistence, keys, and extents, and by defining specific structured literals for date and time support.

In the ODMG C++ binding, types map into C++ classes in a natural way. The template class d_Ref<T> captures the semantics of referring to persistent objects. The primary mechanism is the overloaded -> operator, which does whatever is necessary to return a valid pointer T* from a d_Ref<T>.The d_Ref_Any class provides an untyped reference to a persistent object, for use in such contexts as object naming. Relationships (such as contains/is-contained-by between Runs and Events) are used to guarantee referential integrity; they are implemented as template classes whose interfaces reflect this smart pointer approach (for example, d_Set<d_Ref<T> >). (The distracting d_ prefix was introduced in ODMG-93 Release 1.2 to avoid name collisions.) The C++ binding also introduces standard-length types d_Long, d_Double, and so on, to facilitate operation in heterogeneous environments.

ODMG has also defined a query language (OQL), which has recently been extended to be "a superset of the part of the standard SQL that deals with database queries," with additional features to support object orientation. The C++ binding supports a loosely coupled approach to queries, in which d_OQL_Query objects are constructed from OQL query strings with parameter substitution, and executed via a free-standing function called d_oql_execute.

3.3.2 Related projects in HENP

Recent experience in the physics community (e.g., the PASS project with SSC data models and simulation data; the FNAL Computing for Analysis Project with D0 data; CERN R&D projects for the Large Hadron Collider) has shown that object models are appropriate, and that the ODMG data model in particular is rich enough, to describe the kinds of physics data whose analysis is the target of this project. The R&D projects at CERN (MOOSE and RD45) which the ATLAS collaboration is working with and which related (and complementary to) the proposed work. MOOSE is a study of Object Oriented (OO) programming techniques as applied to HEP event reconstruction systems. RD45 is study of a "Persistent Object Manager for HEP" which is attempting to build data storage and access on "commercial" object oriented data bases. We will provide coordination between the work of the project presented here and the work of MOOSE

and RD45, with the goal of providing a more uniform application interface to storage and access systems for scientific data.

3.3.3 Plans

We plan to build application interfaces to physics data that are compliant with proposed object database standards. While we expect that commercial databases will not, in the time frame of this project, have the performance or parallel processing or hierarchical data access capabilities that we require, we intend to build a system in which migration to commercial products (perhaps one component at a time), or transparent, concurrent use of such products for suitable portions of the data, is entirely feasible.

The Argonne members of this collaboration have, as part of the PASS project, implemented a substantial subset of the ODMG-93 interface as a layer on top of their own persistence software. Their design, though, is intended to provide a layer that is portable to a wide range of underlying object persistence managers; indeed, one of the questions they have addressed is the definition of a minimal functionality that a persistence manager must provide in order that an ODMG-compliant database be buildable above it. We intend to leverage this work in our software development.

It is important to note that there are significant research issues involved both in the design of this interface and in its implementation. The PASS project has, for example, noted a number scalability concerns in the current ODMG C++ standard, including: obstacles to scalability in the specifications for collections and iterators; inadequacies in object clustering specifications; scoping rules (regarding transactions and how long an object must be memory-resident) that significantly impede large-scale data access. We intend to influence the direction of these standards to accommodate the massive high-performance data requirements of the physics community.

On the implementation side, there are equivalent research problems. Among those we plan to address are: instantiations of collections that are maximally scaleable (e.g., by allowing nondeterminacy, disjoint partitioning among variable numbers of processes, and collection nesting); support for object clustering and reclustering; data segmentation and transfer policies (e.g., the appropriate scale for efficient data transfer from hierarchical storage may not match the appropriate scale for efficient retention in memory when an object is touched by a physics query). There are both specification and implementation issues involved regarding data caching and replication policies when multiple queries request access to the same data.

Object oriented databases often support additional capabilities beyond those specified in the ODMG proposed standard. We are particularly interested in schema evolution and versioning as potentially significant tools for effective management of large amounts of data over time, and in efficient and innovative indexing, described elsewhere in this proposal, but also not a part of the ODMG specification.

Effective physical and logical data organization will be critical to the success of this project. This is true of any data store for which the time required simply to read the data is a limiting factor, but it is particularly true of databases that must reside largely on tape---on high capacity devices, touching even an extremely small percentage of the data may mean mounting nearly every tape.

We will support control of data organization on multiple levels, including:

- organization expressed by the physics data models' object definitions;

- clustering at object creation time, as expressed in the ODMG specification (e.g., place a single event's muons "near" one another);
- clustering determined by indexing schemes, possibly transparent to physicists (e.g., put all two-muon events in the same set of tiles);
- management of data segment placement on physical storage devices (examples include concatenation of "consecutive" segments, or round-robin striping, on raw RAID devices or in large Unitree files, distribution across a heterogeneous mix of storage media, and segment replication).

Some storage subsystems may introduce yet another level of physical organization, such as automatic caching on RAID or migration to tape based upon access patterns, as in current Unitree implementations. In conjunction with HPSS researchers and computational infrastructure personnel, we plan to be able both to exploit and to impede such capabilities as appropriate to maximize performance.

The ability to readily reconfigure our physical storage utilization will be an important tool in our work. The Argonne members of this collaboration have, as part of the PASS project, built a lightweight object persistence manager whose design criteria include:

- access to every persistent object from every query node;
- extensible support for a variety of storage mechanisms, including local and remote disk, raw RAID, Unitree file systems, raw device access to DD2 and 8mm tape, parallel file systems such as (formerly) IBM's Vesta and (currently) IBM's PIOFS, and Internet data access via standard FTP and HTTP mechanisms or cgi-bin scripts;
- support for efficient reorganization of data, including segment-level striping, concatenation, and reclustering, without knowledge of object schemata;
- support for data replication;
- support for multiple access paths to data;
- portability to heterogeneous distributed architectures.

We plan to make use of this technology in our work on alternative physical data organization strategies and in our testing of algorithms for multilevel indexing.

Indexing schemes interact at several levels with physical data organization. Some of these, such as the organization of data into tiles and data segment/tile placement based upon multidimensional indexes, are described in our discussion of indexing, but there are many others.

In the physical design of a data store, for example, a natural question might be whether to store the muons associated with a given event with the event object, or in a separate muon store or cluster. Storing the muons with the event has adverse performance implications needless data transfer for queries that do not touch the muons, but storing them separately slows queries that do. A rudimentary quantitative analysis of such tradeoffs, based on query workload characterization, was done as part of the PASS project. This is a particularly important issue for indexing strategies the effectiveness of storing all two-muon events together is mitigated when the muons themselves are not similarly clustered. The exploration of the interaction among these many potential levels of data organization are an important part of our proposed work.

Implicit in the above discussion is the need to provide tools for database instrumentation. We need the ability to monitor system performance, and to decide when and how to reconfigure our

indices and storage utilization. These are needed both for development purposes to measure the effectiveness of our approaches and as deliverable database administration tools, for use when access patterns change, a priori query characterizations prove to be wrong, or new data are introduced.

3.4 Parallel processing for physics analysis

It is characteristic of our data and our physics analyses that queries can often be parallelized almost arbitrarily in principle. We may seek, for example, all events satisfying a certain criterion whose determination may be quite computationally intensive, but those events are often essentially independent, or conditionally independent given experimental run parameters, so that we could analyze one event per processor on an infinitely large parallel machine. There are a number of obstacles, though, to achieving this scalable parallelism. In this section, we describe our plans to accomplish this end.

We propose to build a system capable of exploiting event-level parallelism in a scalable fashion for event reconstruction, DST analysis, and physics analysis.

At the highest level, we will implement a system capable of accepting a query over a large collection of events, disaggregating it into queries running on a variable number of parallel processors, each addressing a disjoint subcollection of events, and aggregating the query output.

It will be important to design and implement collections that support nondeterminism in the order in which elements are returned; without this, there are serialization problems that significantly impede scalability, or else queries must run essentially in batch mode, with a substantial sorting task remaining as a postprocessing step.

We will develop iterators over large collections, whose implementations are transparently parallel--the physicist need not know that elements are being gathered from more than one processor. While transparent parallelism means that the physicist's interface to these facilities does not reveal the underlying parallel implementation, several interface issues nonetheless remain. We plan to use the ODMG's proposed interfaces to unordered collections (the template classes d_Set<T> and d_Bag<T> in the C++ binding) and their corresponding iterators as starting points, while addressing interface requirements in the ODMG standard that may impede scalability (such as the requirement that iterators satisfy Standard Template Library bidirectionality constraints). We must also be able to return partial results while queries execute, a capability not found, for example, in OMDG's d_oql_execute specification.

Achieving scalable parallelism will require much more than parallel implementation of container classes and their iterators. If data were distributed disjointly across a fixed, dedicated set of processors, one might be able simply to replicate a query on each of those processors, and then aggregate the results. (Even in this setting, queries that are too computationally intensive to keep up with I/O rates require additional parallelism.) In large-scale computing facilities, though, computing nodes have direct or indirect shared access to multilevel mass storage. This means that partitioning the query requires significantly more effort--we must decide how many nodes and which nodes run the query, and we must adjudicate which events are processed by which nodes without introducing significant serial bottlenecks. Factors include how the data were organized when the database was populated, proximity to data, load balancing, specifics of the I/O architecture, computational intensiveness of the query, and optimization with respect to other simultaneous queries.

We expect that a great deal of work will be needed, in collaboration with I/O infrastructure personnel, to accomplish this scalable parallelism. Some of these, particularly those related to tape access and control of physical data layout, were mentioned in connection with our planned approaches to indexing. For databases that reside largely on tape, we need to be able to optimize

access for simultaneous queries. We must be able to determine the data needs of concurrent queries and sort (or have the storage system sort) their tape retrievals into ordered per tape requests. (Apart from serious I/O speed degradation, tapes are not random access devices--they break under such utilization.) We would like also for queries to be able to take advantage of data cached on disk from tape by earlier queries.

To query very large databases, we will need the capability of scheduling resources so that we are able to deliver data to compute nodes as fast as data can be read, AND have access to enough computing capacity--dependent upon the particular query--to keep up with data read at this rate. Such capabilities are not found in today's schedulers for massively parallel systems, which tend to allow users to request a specific number of nodes, but not a specific I/O topology. The problem can be particular nettlesome, since the appropriate number of processors for a query may depend upon such details as the accidental assignment of the query to a set of processors---8 processors in a single rack or tower, for example, may be less effective than 4 in multiple racks, because I/O connections to mass storage are often routed through a single distinguished node in each rack.

An example of the potential for subtle dependence on how databases are populated might be in order here. A physicist might run a large-scale simulation by replicating it, with different random number seeds, on each of P parallel processors. It is likely that, to achieve reasonable scalability, these processes will attempt to fill disjoint pieces of persistent storage. The effective consequence may be physical clustering into at least P clusters, where P is entirely an artifact--an accident of the number of processors available when the simulation is run. That artifact will, however, persist in the physical layout of the database, and affect optimal storage access for subsequent queries, no matter which or how many processors those queries may utilize.

We plan to explore collection constructs and implementations with a particular emphasis on scalability. Some approaches, such as ParSets, which explicitly declare the disjoint nature of set contents, have appeared in recent database literature, although there may be some static dependence on how data are allocated. Other high-level approaches will be explored as well. For example, the ODMG specification allows one to say A is the union of B and C, but this is an assignment, not a definition--if one later adds an element to B, it does not become an element of A. Collection constructs that allow A to be defined as the union of B and C enhance scalability. (Imagine a collection All_Events, and what may happen to queries if, in a transaction-based architecture, an exclusive lock must be acquired on All_Events every time an event is added to the database.)

Because of our data scales, it will be possible for physicists to frame queries that may require minutes or months to run. We hope to be able to estimate the cost of a query in our architecture. As a first-order example, we might count the number of tape mounts that a query will initiate; such a capability might be a natural consequence of the kinds of tape system scheduling proposed above.

While our emphasis here has been upon parallelism, it is our intention to deliver an implementation that, from a physicist's point of view, looks the same whether she/he is querying a 10 MB sample on a desktop workstation or a 1 TB sample on a massively parallel processor--the query and the physics code's interface to the data should be identical.

3.5 User Interface and visualization

Visualization tools play an important role in the analysis of Nuclear and High Energy Physics experiments. There are two aspects of data processing where the graphics representation is essential. One of them is the visual interface to the data catalog and the other is a graphical representation of the event reconstruction.

A visual interface to the catalog will provide tools to browse through the data, look for correlations and select events of interest. With this tool, users will be able to see representations of groups of events or single events, perform analysis and retrieve the original, raw, or pre-processed event data from the event database for further studies.

Once the user has selected a sample of events that contain an interesting physics signal, it may be necessary to look at their visual representation. Displaying an event allows visual analysis or detection of inconsistencies that are not otherwise obvious. We propose to provide this capability using an interactive visualization called the Event Viewer. Although an Event Viewer has to be custom-written for each experiment separately, there are many aspects that are common to all the experiments described in this section.

3.5.1 Development challenges of visualization

The users of the system will have varied hardware and network configurations. They will range from very powerful graphics workstations with virtual reality (VR) input/output devices and high-speed networks to low-end desktop machines with very limited graphics capability and slow network connections. This presents many problems due to the highly graphics-intensive nature of these applications. A reasonable minimum configuration that is supported has to be specified.

It will not be feasible in the Event Viewer, due to the very large size of the unpacked data associated with each event (approx. 100 megabytes), to visualize the entire dataset at once. Tasks that become very time consuming under such circumstances include object generation (the process of creating geometric objects that represent the data), rendering (the process of drawing that object on the screen), and data transfer. They require time approximately proportional to the size of the dataset. In general, rendering is the more time-consuming of the two, unless graphics hardware is available. This hardware is becoming more and more common, but at present most low-end workstations must do rendering in software.

In addition to the time requirements, attempt to visualize large amounts of data introduces a problem of information overload. So much information is presented to the user at once that she/he is unable to understand it. To avoid it a reduction of the complexity of a scene such as sampling the data, grouping like objects into a single object, and removing unnecessary objects will be performed. This process of data reduction is common in scientific visualization, and there is a large body of research to draw from in solving this problem.

Some tasks can be offloaded to the graphics server (called "load balancing") if the client is too underpowered to handle the entire job. One advantage of offloading work onto the server is that that the server machine can be connected to the Distributed Parallel Storage System (DPSS)[25] database via a very fast network. This will greatly speed object generation. Rendering tasks can also be performed at great speed on the server, with only the resulting image sent over the network to the client. However, it is advantageous to do as much work as possible on the client, to best make use of all available computing power.

3.5.2 The Event Catalog Viewer

The event catalog contains, for each event from the event database, basic event characteristics such as the physical and detector conditions under which the event was taken, and physics parameters of the event. For example for RHIC experiments, some of the information contained in the event catalog will include the temperature, the total number of particles in the event, the total number of specific types of particles (such as protons, pions and anti-protons), the slope parameter of the event, the width of rapidity distribution in the event, etc. For approximately 20 words per event, the size of the event catalog for one year worth of data will be of the order of at least one gigabyte.

The event catalog visualization tool will give physicists the ability to view the catalog information remotely and locally. It will also have the ability to request a retrieval of the original raw or DST data from the event database. This will allow the users to do further processing and displaying of the raw data based upon the analysis performed on the event catalog.

The ability to retrieve the event catalog data from remote locations using different bandwidth, perform the analysis and still provide the user with reasonably low latency and high feedback rates, will have to be addressed in the development of the tool. Some of the techniques already applied in other domains, such as terrain visualization [26] will be looked at for applicability. The data in the event catalog, with its multiple fields, is a multi-dimensional space going beyond the three dimensions of a terrain database and the terrain visualization approach will have to be significantly modified. Dependent upon the type of analysis being conducted, the retrieval and display of the relevant fields becomes important, especially either over a large database or over a low bandwidth network.

Visual examination of large multi-dimensional databases poses several unique user interface issues. The manipulation of databases in a 3D space on a screen offers several significant advantages. Occlusion, where some of the data is obscured by other data, is addressed by allowing the user to manipulate the data by rotation, translation or scaling. Problems with occlusion and manipulation of various databases have been looked at before[27], however most of these datasets were located locally to the user and fairly small in comparison. How well some of the techniques developed for handling smaller and local databases extends to larger and remotely located databases is an issue which we will address.

The use of stereo displays coupled with a head tracker which allows the user to control the viewpoint has been shown to increase the amount of data on a flat screen that is comprehendable by users. [28] A stereo display gives the illusion of a true three dimensional object by the use of a pair of glasses which gives each eye a separate viewpoint. The fusion of these two images gives the illusion of depth. A tracker, which keeps track of the orientation of the user's head, allows a system to change the viewpoint that is being displayed so that the user has an illusion of moving around the object being displayed. The combination of a stereo display along with a tracker greatly increases the amount of comprehendable data, especially when the data is of a spatial nature.

3.5.3 The Event Viewer

The visualization software will be able to display all available types of data, including raw data, reconstructed information, and simulated data and color the objects according to multiple independent variables. It will integrate data reduction, on-demand fetching, and load balancing to maximize frame rate while maintaining sufficient information content for visual analysis or anomaly identification. Event Viewer will be similar to prototypes written by STAR collaborators.

The main goal of the Event Viewer is to enable the researcher to quickly perform a visual analysis of the event or determine if there was anything wrong with the detector during the event. To that end, the user will be presented with a series of representations of the data, each at a different level of detail. The initial view will be a global picture of the whole detector at a very low level of detail (LOD). This level will give enough information to provide an event overview. As the user zooms in on parts of the detector that are of interest, the detail will be increased incrementally until the user is satisfied and stops, or the highest LOD is reached. At each level, the region of interest must be significantly smaller than at the previous level in order to avoid degrading performance unacceptably.

We will investigate ways to avoid excessive network delays and take advantage of the features of the DPSS by retrieving data on an on-demand basis. That is, only the data that is necessary for the current LOD will be fetched. We will also look into caching data locally.

3.6 Interface to experiments

There are three general aspects where each of the experiments contribute to this project. The first is in providing data (either simulated data or, in the case of CLAS, real detector derived data), the second is the interface between the experiment specific format of data as it comes from the detector and the object model applications program interface (API) of this project, and the third is in using the system which is developed for actual data mining, simulations and analysis. At the detail level, there are differences of what each experiment will contribute and the level of effort involved. The particular areas of involvement and basic motivation for each experiment is listed below.

- ATLAS - This GCA is viewed as a large scale prototype for ATLAS and the involvement is focused upon developing a prototype consistent with the corresponding CERN effort (RD45). The result of this prototype should be an understanding of how to scale up to the size of the event reconstructed data (1 PB/year) and the number of physicists (1200). Immediate use will be made by US-ATLAS of the system to carry out simulation studies.
- BABAR - This GCA is the same scale as BABAR requirements and it is complementary to the on-going developments of the BABAR project at SLAC and their collaborating institutions. The BABAR effort on data representation and data access is consistent with the RD45 project and the participation of BABAR will contribute greatly in this area. In addition , this project will facilitate the design of the most effective data analysis system for BABAR.
- CLAS - This is the single experiment which will have any significant amount of real detector data within the three year time scale of the project since it is scheduled to begin in 1997. The issue of storing the data according to how it will be accessed rather than efficient random access is most critical for CLAS. This aspect fits in well with the overall GCA schedule in which the efficient indexing for random access is one of the later goals compared to the capability to layout the storage according to the expected access pattern.
- PHENIX, STAR - These are the two largest RHIC experiments. They already started the Monte Carlo data production this year and for successful data analysis they need an amount of simulated data equal to a yearly production of real data. By the end of 1996 these experiments will provide enough data to develop and test those aspects of GCA that have a direct application in developing the necessary tools for efficient data access and most effective remote data analysis for RHIC physics program.

We will use the LBNL/NERSC and ANL/CCST centers interconnected by the ESnet WAN as a model test-bed for a distributed regional data access centers which would fit the computing and analysis model being planned for the experiments (.See Figure [http://sun2.hep.anl.gov/EdMay.Myweb/gca.ps] for ATLAS). The hardware to be used at ANL is summarized in table 3.1. Simulated data generated at LBNL will be bulk loaded into the Hierarchical Storage Manager (HSM). This data will be converted into an object-oriented data model[29] (at the coarse level: raw data objects, reconstructed objects, physics data objects, statistical distribution and physics summary objects) and redistributed over the HSM and parallel file systems at the LBNL and ANL centers. Smart query tools and data base like access would

be provided for researchers to analyze and visualize the stored object data. In addition, the analysis system will be installed at BNL for use by the RHIC experiments.

Table 3.1

	Counts			
	FY97	FY98	FY99	Total
HPC Cycles (GFLOPS hours)	10^5	5.10^5	10^6	$1.5x10^6$
HSM storage (Tbytes)	1	5	10	16
I/O HSM to MMP memory (GB day)	100	1000	5000	
Visualization (hours of CAVE/IDESK time/month)	small			
Collaborative tools (# of desktops * hours/week)				2 x 20
Network access (Mbits/site/month)	OC3 ATM links to ANL-HEP (dedicated) OC3 or OC12 link ANL-LBNL (10^5 - 10^6)			

4. Goals/deliverables

The ultimate goal of this GCA is to provide future experiments with a reliable system with a quantum leap in performance which will enable scientists to analyze massive data sets in a timely fashion. This will require new, innovative solutions to outstanding technical problems in the field. In order to meet the experimental requirements, a significant number of computer-related deliverables have to be provided (see below). Even partial achievement of these goals will represent a major milestone on the path toward providing essential tools for the HENP research program. The list of computer-related goals/deliverables follows:

1. Data Storage (MSS) - hierarchical storage of 50 TB.
2. Processing farm - array of SMP machines with an aggregate peak performance of 100 GFLOPS.
3. Persistent object system - C++ object storage and retrieval software compatible with the ODMG-93 API specification and permitting interface to experiment specific data formats.
4. Optimization of data storage and access - software integrated with the persistent object system (3.) which will optimize the storage and access of data based upon predicted access patterns, and also allow reorganization of the data storage based upon different access patterns.
5. Parallel event processing system - software framework for executing physics analysis algorithms in an event-parallel fashion, accessing data via (3.), aggregating results for physics summaries and visualization, and provide performance statistics (number of events processed, number of events lost, etc.
6. User interface - user interface to data browsing (data catalog), data query, event processing and visualization.

7. Visualization - display of physics summaries (histograms, ntuples, …) locally and across LAN and WAN. High-performance display of multi-dimensional data. Includes C++ class library for physics summary data objects and software to display of these data objects
8. Experiment interface - each experiment will use the interface provided in (3.) to connect with their own experiment specific data.
9. Simulations - experiments will generate simulated data on the NERSC facility.

The performance goals are:

1. LAN Data Access - access to MSS from farm with 100 MB/sec bandwidth at NERSC.
2. Desktop access - reliable delivery of 1 GB/day between desktop computers across the US (at ESnet sites) and a DFS file server.
3. WAN data access - reliable transport of 1 TB data sets between centers (LBNL, ANL, BNL, CEBAF) over ESnet at a rate of 1 TB/week.
4. Event parallel processing - execution of event parallel processing with an aggregate peak of 100 GFLOPS and 100 MB/sec.
5. Query - query of 50 TB of data to produce 1 TB selection.
6. Monitoring - monitoring to determine the efficiency of data access and parallel processing

The demonstrations of performance goals and portability of the software are:

1. All performance goals will demonstrated at NERSC.
2. Deliverables 3, 4, 5, 6, 7 and a subset of 8 will be demonstrated at BNL and ANL.

5. Work plan

5.1 Management Structure

The PI's will call biweekly video meetings, and quarterly full group meetings. At these meetings, the progress towards the deliverables listed in section four will be reviewed. An annual written summary of progress will be distributed. Responsibilities of each institution are outlined below.

Integration of the product of this GCA into the ongoing HENP program is assured since we will install the system at ANL and BNL as well as NERSC. Further, the physicist members of the collaboration are active participants from HENP experiments involved.

5.2 Summary of work at each site

The following list describes the work to be carried out by each of the participating institutions:

ANL	ODMG programming API
	Object model for ATLAS
	System performance evaluation for ATLAS case
	Test system at ANL
	Contribute to parallel event processing

BNL	Interface to hierarchical mass store to ensure portability to RHIC system System modeling of this GCA system Transport of TB datasets between centers
FSU	Programming interface to CLAS data Provide CLAS data Storage organization based upon event features CLAS analysis software
LBNL-STAR	Programming interface to STAR data Provide STAR data (simulations) High-performance visualization Parallel event processing Project management
LBNL-ATLAS	Provide ATLAS data (simulations) Project management
LBNL-NERSC	Optimization of storage organization and access Interface to NSL-Unitree (and HPSS) ODMG programming API Parallel event processing High-performance visualization Transport of TB datasets between centers Performance monitoring Hardware installation & operation System support for PDSF, MSS, DFS, DPSS
UCLA	Programming interface to STAR data Test site for wide area desktop access Provide STAR data (simulations) Remote visualization
U. Tenn.	Programming interface to PHENIX data Provide PHENIX data (simulations) Parallel event processing Event visualization
Yale	Adaptation of PHENIX data structures Provide PHENIX data (simulations) Test site for wide area desktop access Performance evaluations

5.3 Schedule

5/96 - 6/96 Write proposal

6/96 - Submit Proposal

6/96 - 9/96 - Prepare requirements of experiments

10/96 - 12/96 - Prepare detailed project plan

1/97 - 3/97 - Scope technical choice projects

4/97 - 9/97 - Evaluate & test technical choices

8/97 - 9/97 - Update plan for phase I and phase II

9/97 - Review status and plans

10/97 - 3/98 - Implement phase I

4/98 - 9/98 - Test and evaluate phase I

8-98 - 9/98 - Update plan for phase II

9/98 - Review status and plans

10/98 - 3/99 Implement phase II

4/99 - 9/99 Operate phase II in production

9/99 - Final project review

6. Budget

The following tables summarize the labor and equipment needs for this GCA. The persons named in the column marked 'person' are either the actual individual who will carry out the work, or the coordinator for others. In some cases, the names of the persons to be carried on project funds have not yet been determined. The column marked "continuing" refers to a matching effort from ongoing operation support. Labor costs are estimated at the full overhead rate as appropriate for each institution. Note , that much of the software development requires extension of existing work rather than development of all new systems from "scratch." We estimate that the proposed manpower will be capable of producing deliverables listed in section 4. No manpower contingency is applied since we believe additional continuing support can be found as needed by extending the collaboration to include other institutions.

Cost for the major equipment items listed in Table 6.3 are estimated from vendor quotes or by extrapolation from current costs. Other than the ANL equipment, all the major items will be installed at LBNL. The performance goals for each fiscal year are as shown, with breakdown in the case of CPU peak Gflop into PDSF and T3E time. No overhead has been included in these estimates. A 20% contingency has been included in estimated equipment costs. Table 6.4 shows labor costs and summarizes the total project cost. All costs are quoted in the FY96 dollars.

Table 6.1: Scientific Support FTE's

Institute	Person	Group	proj.	Continuing
LBNL	D. Olson, G. Odyniec, I. Sakrejda	NSD, STAR	1.5	1.5
LBNL	J. Siegrist, I. Hinchliffe	Physics, ATLAS	0.5	0.5
UCLA	H. Huang	Physics, STAR	0.5	0.5
U. Tenn	S. Sorensen	JINR, PHENIX	1	1
Yale	S. Kumar	Physics, PHENIX	0.5	1
FSU	G. Riccardi	CS, CLAS	0.5	0.5
BNL	B. Gibbard	RHICC	1	1
ANL	E. May, D. Malon	Physics, ATLAS	1	1
		Totals	6.5	7

Table 6.2: Infrastructure Support FTE's

Institute	Person	Group	proj.	Continuing
LBNL	Culler	NERSC	0.5	
LBNL	Barber	NERSC		1
LBNL	NP consultant	RNC, STAR		0.5
LBNL	HEP consultant	Physics, ATLAS		0.5
LBNL	A. Shoshani	ICSD	1	
LBNL	T. Welcome	NERSC	0.5	
LBNL	N. Johnston	ICSD	0.5	
LBNL	W. Johnston	ICSD	0.5	
		Totals	3	2

Table 6.3: Major NERSC Equipment Items - Hardware & Software

ITEM	FY97 Performance Goal	FY97 Cost($K)	FY98 Goal	FY98 Cost($K)	FY99 Goal	FY99 cost($K)
CPU						
PDSF	5	500	20	1000	100	2000
T3E	5	0	5	0	5	0
Cumulative Total	10 Gflop peak		25 Gflop peak		100 Gflop peak	
Robot Mass Store	10 TB	251	50 TB	600	50TB	0
Disk	1 TB	200	3 TB	400	10 TB	700
Software Costs		50		50		50
Network Hardware		100		100		50
Contingency		0.2		0.2		0.2
Totals		1321		2580		3360

Table 6.4: Project Personnel Cost and Cost Summary

Project Personnel Costs	FTE	K$
Scientific Support	6.5	1207
Infrastructure Support	3	621
Travel (4 trips/yr/FTE)		68
M&S costs		90
	Total	1986

Cost Summary	FY97 ($K)	FY98 ($K)	FY99 ($K)
LBNL Totals	3307	4556	5346
ANL equipment	957	957	957

7. Evaluation Criteria

7.1 Fundamental Significance

Two most fundamental questions facing nuclear and high energy physics today, namely characterization of the transition to the Quark-Gluon Plasma (QGP) phase of matter and the discovery of the mechanism responsible for electroweak symmetry breaking, require advances in computational capabilities, information management, and multi-user data access.

7.2 DOE Mission

The office of HENP at DOE has committed enormous resources to the construction of new accelerators and major detectors. The return on these investments will depend critically on our ability to capture, store, access, and analyze the massive data sets that will be generated. If we fail to meet these unprecedented challenges, some, perhaps much, of the larger effort will be wasted.

7.3 HPCC Goals

Technology:
Analyzing HENP data at 50 TB level poses several technology challenges. This GCA will develop and apply enabling technologies that will demonstrate a high-performance persistent object database capable of referencing 1PB of data consisting of 10^{12} or more objects. This database will use a hierarchical storage system with the lowest level of storage residing on tape. Monitoring the performance of this database will be integral to its successful operation. While this database technology will be suited to data analysis for high-energy and nuclear physics, the indexing and storage techniques will be of significant importance to general purpose very large databases.

Education:
The high-energy and nuclear physics programs in DOE are integrally related to U.S. higher education in the physical sciences. Most of the collaborating institutions of the experiments

participating in this GCA (as well as HENP in general) are universities. Participation in these experiments is an essential aspect of the graduate work for hundreds of students. A primary result of this successful GCA project will enable a much greater degree of hands-on activity for these students to understand, analyze, and derive significant physics results from the data in these experiments, and benefit from computer science technology developed in this project.

7.4 Enabling Technologies

A number of developments in computing hardware and software will provide the enabling technologies to address this Grand Challenge problem. Multilevel storage systems with the necessary capacity and performance are within reach, and parallel computing platforms with the computing power to support the analysis of such massive amounts of data will also become available.

The proposed work matches the timeline of expected advances in enabling software. It is clearly a project for a post Unitree storage management environment, one that will both exploit and help set the agenda for HPSS research and development. It is a project that can help define the requirements of the next generation of parallel computing system schedulers, requiring a level of matching to I/O that is beyond today's "any N nodes for H hours" scheduling. It is a project well positioned to take advantage of developments in the scalable I/O initiative. The maturation of object technologies and the emergence of object database standards make this a timely project, in that they reduce the risk of producing homegrown solutions that do not allow ready migration to commercial technologies as they become available. Finally, the work is well positioned both in time and in project personnel to take advantage of expertise and software developed in earlier initiatives such as MAGIC and PASS.

7.5 Interdisciplinary Approach

To achieve the goals of this proposal it is necessary that physicists and computer scientists cooperate to develop jointly the various components of the system. The team assembled was specifically chosen to have expertise in relevant areas. This proposal was developed by identifying the bottlenecks that require such expertise. The proposed solutions include research, development, and deployment of the system components. Physicists will provide the data, the analysis requirements, and the domain knowledge, and computer scientists will provide the infrastructure, algorithms for efficient processing and storage access, visualization, and the software technology. We expect these disciplines to interact iteratively, testing prototype results, developing joint incremental inprovements, and robust products. All the collaborators have participated in or lead projects relevant to the tasks identified; they will be able to build on their experience and knowledge. The expertise in the physics area includes reconstruction and analysis of high energy and nuclear physics events. The expertise in the computer science area includes: object-oriented software development (at LBNL, and ANL), high performance parallel computation (at NERSC, FSU, U. Tenn), efficient access of mass storage systems (at LBNL), and visualization (at LBNL).

7.6 Support Leveraging

As listed in Table 6.1, the research component of this project includes 6.5 FTE paid by the GCA and 7 FTE paid by continuing program funds.

7.7 Technology Leveraging

A transformation of scientific computing has begun that has caused "data intensive computing" to become an increasingly common theme in discussions of the future of the nations high performance computing infrastructure. For example, the San Diego Supercomputer Center has chosen data intensive computing as the major theme for its proposal in the National Science Foundation recompetition of its supercomputer centers. Because storage and data management issues have been over shadowed by the the focus on increasing processor capability over the past decade, it is widely recognized that we have reached a juncture where for many of our most important grand challenges in computing the bottleneck has become data management.

Global climate simulation is such a case, where decades or centuries of detailed information can be created in single simulations which produce 40 TB of information. Analyzing and mining that data is an essential part of that research, and since the basic technical underpinnings of that analysis are currently ill-formed at best, that discipline is focussing on the problem. Analysis of the data from the Earth Orbiting Satellite data bases is another example, as are applications in genomics (where data fusion with information on macromolecular structure is important) and in analysis of astronomical observations, where the ability to collect digital images using charge coupled devices (CCDs) has transformed the nature of data collection.

Efforts in the other DOE grand challenges involving data mining (finding correlations across huge data sets), and link distributed large data sets with predictive models will benefit directly from the persistent object software development we will do in this grand challenge. Parallel I/O systems and a uniform method of accessing data in the (physically distributed) data archives is also required in many of these applications, and that is precisely what is being developed here.

Perhaps the most important part of performing this research in the context of a GCA collaborating with NERSC is that this mode of development will accelerate the adoption of the technology we develop in other disciplines and will allow us to leverage their results as well.

7.8 Computer Resources

The HENP computing problem is well suited to the proposed NERSC infrastructure upgrades. Storage of about 50TB and 100 peak GFLOPS of processing capacity match well requirements of future HENP experiments.

7.9 Multiple Platforms

Software portability will be demonstrated by installing and running the system at NERSC, ANL, and BNL.

8. Glossary of Acronyms

ANL	Argonne National Laboratory
API	Application Program Interface
ATLAS	HEP experiment at LHC

ATM	Asynchronous Transfer Mode
BABAR	HEP experiment at SLAC
CDF	HEP experiment at Fermilab
CEBAF	Continuous Electron Beam Accelerator Facility
CERN	Center for European Nuclear Research
CLAS	NP experiment at CEBAF
D0	HEP experiment at Fermilab
DFS	Distributed File System
DPSS	Distributed Parallel Storage System
DST	Data Summary Tape
FNAL	Fermi National Accelerator Laboratory
GB	gigabyte
GCA	Grand Challenge Application
GFLOP	Giga Floating Point Operations
HENP	High Energy and Nuclear Physics divisions of DOE
HEP	High Energy Physics
HPPI	High Performance Packet Interconnect
HPSS	High Performance Storage System
HSM	Hierarchical Storage Manager
LAN	Local Area computer Network
LBNL	Lawrence Berkeley National Laboratory
LHC	Large Hadron Collider, at CERN
MSS	Mass Storage System
NERSC	National Energy Research Supercomputer Center
NP	Nuclear Physics
NSL	National Storage Laboratory
ODL	Object Definition Language
ODMG	Object Database Management Group
OO	Object Oriented
OQL	Object Query Language
PDSF	Parallel Distributed Supercomputer Facility
PHENIX	NP experiment at RHIC
QCD	Quantum ChromoDynamics

QGP	Quark-Gluon Plasma
RHIC	Relativistic Heavy Ion Collider, at BNL
SLAC	Stanford Linear Accelerator Center
STAR	NP experiment at RHIC
SUN SMP	Sun's Symmetric MultiProcessor
TB	terabyte
TJNAF	Thomas Jefferson National Accelerator Facility, the new name for CEBAF
VR	Virtual Reality
WAN	Wide Area computer Network

---

[1] Long Range Plan for Nuclear Science DOE/NSF(1995)

[2] See Glossary of Acronyms

[3] The CDF Detector: An Overview F.Abe et al. Nucl. Instr. Meth A271 (1988) 387.

[4] The D0 Detector Nucl. Instr. Meth A338 (1994) 185.

[5] Parallel query processing for event store data. A multilevel object store and its application to HEP data analysis. Proceedings of Computing in High Energy Physics 1994, edited by S.C. Loken, pp 229-240, 1995.PASS

[6] http://www.cn.cern.ch/pl/cernlib/rd45/

[7] http://fnhppc.fnal.gov/cap/cap.html

[8] http://www.rhic.bnl.gov/star/starlib/doc/www/welcome_star.html

[9] H. Satz. Ann. Rev. Part. Nucl. Sci 35 (1985) 245.

[10] J.Collins and M.J.Perry, Phys. Rev. Lett.

[11] J.W. Harris and the STAR Collaboration, "Conceptual Design Report for the Solenoidal Tracker at RHIC". LBL Pub-5347 (1992)

[12] PHENIX Conceptual Design Report, (1993).

[13] http://www.cebaf.gov/clas/CLAS.html/

[14] http://www.cebaf.gov/

[15] "ATLAS: Technical Proposal for General-Purpose pp Experiment at the Large Hadron Collider at CERN", prepared by Atlas Collaboration, CERN/LHCC/94-43, unpublished CERN report 15 Dec 1994.

[16] ROCOCO2 Report

[17] CHEP94 reference to R. Brun

[18] http://www.transarc.com:80/afs/transarc.com/public/www/Public/ProdServ/Product/DFS/dfsoverview.html, http://www.transarc.com:80/afs/transarc.com/public/www/Public/ProdServ/Solutions/larc.html

[19] Chen, T., and D. Rotem, Declustering objects for Visualization, in Proceedings of VLDB 1993, pp. 85-96, Dublin, Ireland.

[20] Chen, T., and D. Rotem, Optimizing Storage of Objects on Mass Storage Systems with Robotic Devices, in proceedings of EDBT (Extending Database technology) 94, Cambridge, U.K.

[21] Chen, T. , R. Drach, M. Keating, S. Louis, D. Rotem, A. Shoshani, Efficient Organization and Access of Multi-Dimensional Datasets on Tertiary Storage Systems, in special issue on Scientific Databases, Information Systems Journal, Pergammon Press, April, 1995.

[22] Chen, T., R. Drach, M. Keating, S. Louis, D. Rotem, A. Shoshani, Optimizing Tertiary Storage Organization and Access for Spatio-Temporal Datasets, NASA Goddard Conference on Mass Storage Systems, March 1995.

[23] Tierney, B. , B. Johnston, T. Chen, H. Herzog, G. Hoo, G. Jin, J. Lee and D. Rotem, "Distributed Parallel Data Storage Systems: A Scalable Approach to High Speed Image Servers", ACM Multimedia, October 1994.

[24] "The Object Database Standard: ODMG-93", Release 1.2, R.G.G. Cattell ed., 1996, Morgan Kaufmann Publishers, San Francisco, ISBN 1-55860-396-4

[25]"A Distributed Parallel Storage Architecture and its Potential Application Within EOSDIS", W. Johnston, B. Tierney, J. Feuquay, T. Butzer. NASA Mass Storage Conference, Goddard Space Flight Center, April, 1995.

[26] Leclerc, Y.G. and Lau, S.Q., Jr., "TerraVision: A TerrainVisualization System", SRIInternational Technical Note #540, Menlo Park, CA 1994

[27] Chuah, M.C., Roth, S.F., Mattis, J., and Kolojejchick, J.,"SDM:Selective Dynamic Manipulation of Visualizations", Proceedings Symposium on User Interface Software and Technology, ACM, 1995

[28] Ware, C. and Franck, G., "Evaluating Stereo and Motion Cues for Visualizing Information Nets in Three Dimensions", ACM Transaction on Graphics, 1996

[29] "Flexible Storage Services for Parallel Data Mining", D.Malon and E.May, "On Persistence Interfaces for Scientific Data Stores", D.Malon and E.May available from http://sun2.hep.anl.gov/PASS.Myweb/index.html