



Fermi National Accelerator Laboratory

FERMILAB-Conf-95/358

**Data Handling and Post-Reconstruction Analysis
at Next Generation Experiments**

M. Fischler and S. Lammel

*Fermi National Accelerator Laboratory
P.O. Box 500, Batavia, Illinois 60510*

November 1995

Proceedings of the *Computing in High Energy Physics 1995 (CHEP '95)*,
Rio de Janeiro, Brazil, September 18-22, 1995

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DATA HANDLING AND POST-RECONSTRUCTION ANALYSIS AT NEXT GENERATION HEP EXPERIMENTS

M. FISCHLER, S. LAMMEL

Fermilab, Batavia, IL 60510, U.S.A.

Abstract: A new generation of experiments in high energy physics is approaching. With the approval of the LHC at CERN and the revised Main Injector project at Fermilab, high statistics experiments will start operation within 5 to 10 years. With luminosities up to $10^{34}/\text{cm}^2/\text{sec}$ and several hundred thousand readout channels, data most likely cannot be handled and analysed using traditional HEP approaches. The CAP group at Fermilab is investigating different approaches to data handling and organization for post-reconstruction analysis. We discuss the approaches considered, their strengths and weaknesses, integration with hierarchical storage, and sharing of primary data resources.

1 Approaches to Post-Reconstruction Analysis

The volume of data involved in High Energy Physics experiments will be increasing by a factor of at least twenty within the next 5–10 years. The prospects of applying current analysis methods, relying on improved computing hardware to cope with the increased flood of data, are marginal at best. Bottlenecks, particularly in bandwidths to disk and tape storage, will have large adverse impacts. Current systems are even marginal for present analysis needs: Physics questions requiring scans of large collider data sets already take weeks to answer; presumably, some investigations which would be more difficult are not being pursued at all.

Approaches to analysis can be characterised by the nature of investigations supported efficiently, and the data and results delivered. Finite computing resources necessitate focusing on a few approaches, providing systems which deal well with large data sets in those contexts. The current analysis approach implicitly chooses a focus driven by realities of yesterday's systems: Easy access to the full data of arbitrary events is sacrificed. The goals of an approach specify what will be convenient and efficient, and might include: cutting on and plotting event attributes; applying complicated selection criteria to access the full data of a small set of events; and applying sequences of simple cuts to refine event sets. It is hard to provide efficient implementations of all goals at once, but integrated support for several approaches provides more versatility than a system tuned for only one.

Currently, experiments organize raw data in banks handled by FORTRAN memory management packages such as ZEBRA and YBOS.¹ The same packages manage banks of information of interest for physics analysis, derived by a compute-intensive reconstruction pass. The "full event data" is kept in a collection of files.

In a second step, events are sorted by characteristics such as existence of energetic electron candidates, and infrequently used raw or intermediate data is dropped. This makes possible early data-intensive analysis, in which the data remains organized via the same memory management packages. Typically, specific information per event is then collected and saved in n-tuples. This information is rapidly available for histogramming and visualization; but access to information other than the

collected attributes can be very time consuming and troublesome. For the current collider experiments at Fermilab, early data-intensive steps have become a significant effort limiting analyses. Since the volume of data will increase faster than storage capacity and bandwidth technology improves, this approach will be even more limiting for the next generation of experiments.

In contrast, several modern approaches choose some model of what the user will want to do, and attempt to achieve those goals in a more efficient and convenient way. One approach, already implemented and popular, is that of PAW/PIAF². The experiment identifies those portions of information which users will want to access—and thus which are worth organizing into n-tuples—and investigations consist of combinations of simple cuts, leading to histograms of selections of the n-tuple data. PAW and PIAF focus on providing a convenient interface for expressing cuts, and on optimizing the accumulation of histograms.

The CAP project at Fermilab assumes that selections may be complex and may involve a larger subset of event attributes, and that full reconstruction summary data or even the entire event may be needed for selected events. The small set of chosen events will be subject to further analysis at user workstations. Selection is made efficient by organizing event data into physics objects (electrons, jets, etc.)—most criteria involve only a few object types so the entire data need not be scanned.

Another approach integrates data management from early in the analysis cycle, down to detailed study of key events on local workstations. This integrated approach assumes that selection cuts are simple and change in incremental fashion, but that the user ultimately needs access to the full data of selected events. The data is organized to optimize the anticipated typical analysis cycle.

Early physics analysis needs high-performance access to large datasets. Each of these approaches features different strengths: convenient access to full data; efficiency in scanning and selection; complexity and flexibility of criteria supported; and/or optimal production of histograms and plots. No single approach is “right”—users analyzing different aspects of the physics will benefit from various capabilities.

2 Hardware Configuration Considerations

Although needs depend on the approaches supported, volume of data, and usage patterns, systems for data-intensive analysis should meet two common requirements: scalability to support usage increases without developing a new system; and connectivity to deliver high bandwidths to each CPU.

Any analysis system will need a large body of data on disk—hundreds of Gbytes of Summary Data for today’s collider experiments, at about \$300/Gbyte. Some approaches can benefit from disk data parallelism, to focus extreme performance on individual queries; this parallelism can be integrated into the approach, or provided by a vendor-supplied parallel file system. An example of the latter is IBM’s VESTA/pfs, which early CAP efforts³ explored in the data-mining context.

Since the full data for experiments is measured in Terabytes, exclusive reliance on disk-resident data is infeasible. Huge repositories of operator-mounted tapes carry the lowest naive price per byte stored, with per Gbyte costs of \$1.10 for data-grade Exabytes, \$3.50 for DLT, or \$4.80 for IBM 3590 cartridges. Here pure media

prices are low enough to be moot, but costs of servicing mount requests—personnel, mount latency, drive breakage and mis-mounts—are devastating.

These costs of human-serviced tape mounts make automated tape libraries (ATLs) attractive. Each robotic library can handle thousands of tapes, and mount more than 3000 tapes a day. The hardware cost is dominated by space for tapes in large, reliable robots. The CAP system will have a library of 2800 10 Gbyte-cartridges—the cost (including robot, drives, media, and serving computers) is \$27/Gbyte. To be useful, an ATL must be controlled by a hierarchical storage management (HSM) system, which supports a model of user access to the data. Good HSM software is not easily available and will be expensive; but the cost is small when amortized over tens of Tbytes of ATL storage. Different approaches can take advantage of robotic tape resources to varying extents.

3 Analysis Methods and Data Organization

The philosophy of an approach—the assumptions made about access patterns and goals—is implemented by organizing data to allow user queries to proceed efficiently. Tricks employed may include isolating data such that typical questions can be answered without bringing in extraneous data; and creating multiple data copies with orthogonal organization.

The PAW/PIAF approach recognizes that n-tuples are a powerful tool for eliminating unnecessary data access and expressing data selection and extraction needs. A fraction of the event data is put into n-tuple format (and kept on disk)—physics thought goes into deciding which information is “hot”, and how it is to be sorted and compactified for optimal access. PIAF pays particular attention to caching frequently accessed data, and to efficiently forming histograms.

The data organization is that of long vectors of single attributes, for example, a vector of electron energies. A query may need to scan only a few of these, to both select and histogram requested data. If few vectors are needed for a query, performance can be excellent without “striping” to parallelize disk access.

PAW/PIAF supports a simple “query language” to facilitate expression of selection criteria, and also supports user-written FORTRAN selection routines. As long as a question requires no data outside the pre-determined n-tuples, it can be expressed conveniently and processed rapidly. But access to data outside the n-tuples remains awkward, and attributes of variable-length are not well supported. PIAF optimizes for an analysis style requiring instant access to a small number of attributes. Other, “othogonal” approaches are complementary and valuable.

CAP goals stem from focusing on situations where limitations imposed by n-tuple methods impede analysis. The assumptions are: some users need data which is too large to keep on disk for every event; selected results will be a tiny fraction of the data set, suitable for scrutiny on a workstation; and selection criteria will range from simple cuts to complex conditions based on multiple physics objects. Data involved in selection is kept on disk; scanning large amounts of data requires parallel access to many disks in a scalable system. Both smooth access to data

on tape (for extraction of full events) and rapid assembly of disk-resident data are required. Details of CAP data flow are presented elsewhere at this conference.⁴

Data is organized so as to avoid excessive input when evaluating criteria, yet minimize fragmentation of disk-resident data making up each event. To do this, the hierarchical structure intrinsic to physics data is translated to C++ definitions of “physics objects. For example, the experiment’s data model defines a “muon” object containing the attributes of a muons, and an event may contain any number of muon objects. C++ objects allow natural support for concepts such as variable-length attributes and pointers “linking” one object to another. A “persistent object manager”⁵ permits access to the physics objects almost as easily as to memory-resident variables, and allows organization into stores of muons, jets, etc. Most of the event summary data is organized by physics object on disk, available for use in selection criteria. Since even complex queries typically involve few types of particles, only a fraction of the disk-resident data need be scanned for a given selection, and stores are striped across several disks to be read in parallel by multiple I/O servers⁵ to optimize performance. Further speed can be achieved by isolating commonly used “popular attributes” into smaller stores. Access is provided to full event data on tape; selected events can be addressed by persistent pointer as if they were on disk or in memory. Frequently requested “hot events” will be cached on disk.

CAP provides an Event Query Language (EQL) which extends the familiar PAW style, automating conditions involving multiple physics objects and links between objects. Although user-defined selection functions are supported, the intent is to make EQL flexible enough to support any query desired. The user can select output in various formats, (native ZEBRA/YBOS, n-tuples of specified attributes, and/or C++ objects), view statistics on rejection by each criterion, and obtain plots of attributes of events passing each stage of cuts.

At the next stage of CAP development, other approaches will be supported on the same system, preferably sharing access to the same data.

In the **Integrated Approach (I-A)**, information at each step, from reconstruction through analysis results, is collected in a (distributed) system of databases. Data organization spans levels of workgroup servers and analysis platforms—each maintains its own database, retrieving information from a primary repository. Specific fractions of the data propagate to platforms far from the primary repository. The approach unifies both the types of data produced, and the set of storage and CPU systems involved in analysis. Communication between platforms is critical.

Any given analysis step usually examines a tiny fraction of most events—but for a very few events, all data should be accessible. A goal is transparent data transport supporting this. High efficiency is based on assumptions about analysis efforts: Activities spend long lifetimes rarely wandering across local platforms; physicists tighten selection cuts more often than loosening them; cuts seldom change to depend on different attributes. The primary data repository is organized by physics object. Disk storage acts as cache, with infrequently used data remaining in an ATL. When a query is posed, relevant data propagates transparently to the user’s analysis platform. Workgroup servers are an intermediate step: A server with high bandwidth to a group of workstations doing related analysis can intercept most

data retrievals before they involve the primary repository.

The objects stored at local levels contain only specific attributes needed for analysis on the local platforms. Tightening cuts involves only data in that database and is very fast. Cut attributes are retained locally for all events, so slight relaxations of criteria are also efficient. Tunable caching schemes retain information at analysis clusters, workstations, and tape storage at various levels. Access patterns can be monitored to create groupings of frequently accessed data.

Thus while PAW is concerned with data organization on disk, and forms n-tuples for efficiency, and CAP organizes stores of physics objects on disk and ATL tape, still within a central system, I-A also addresses hierarchies of partial data sets spanning geographically dispersed systems on a network, and dynamically creates additional organizations when justified by access patterns.

4 Coexistence and Data Sharing

Supporting several approaches allows physicists to choose the appropriate tool for each investigation. If, rather than providing disjoint implementations, we can unify data formats and organizations, then multiple approaches can share one copy of the large bodies of data on disk or tape. But sharing data may involve compromises between efficiency and space, and may be impractical in some cases.

Data in old-fashioned HEP format is not directly usable by the approaches discussed above. Neither the I-A nor the CAP approach can make direct use of the n-tuple data needed by PAW/PIAF, but both can easily generate n-tuple output and would thus be ideal to load such a system.

Both I-A and CAP maintain full datasets in a disk/tape primary store, with event information organized by physics object—they can share this. But when I-A dynamically creates reorganized collections of data, CAP cannot take advantage of them. Intermediate data managed by I-A on workgroup servers and local workstations is not easily shared with the other approaches.

References

1. *ZEBRA Reference Manual*, CERN Program Library Long Writeup Q100/Q101; *YBOS Programmer's Reference Manual*, Fermilab CDF note 156.
2. *PAW—The Complete Reference* and *PAW++ User's Guide*, CERN Program Library Long Writeup Q121, Applications Software Group, CERN (1993).
3. K. Denisenko, E. Paiva, M. Isely, M. Miranda, *Vesta Experiments at Fermilab* Fermilab/IBM CRADA Report, July, 1994.
4. M. Isely, K. Fidler, *et. al.* *The Computing for Analysis Project—A High Performance Physics Analysis System*, Presented at CHEP 95, Rio de Janeiro, Brazil, September 1995. (Proceedings in this volume.)
5. M. Fischler, M. Isely, A. Nigri and F. Rinaldo, *POPM: A Distributed Query System For High Performance Analysis of Very Large Persistent Object Stores*, submitted to IEEE Hawaii International Conference on System Sciences (Jan 1996).