

## Off-line computing for experimental high-energy physics

Richard P Mount

California Institute of Technology, Pasadena, California 91125, USA

### Abstract

The needs of experimental high-energy physics for large-scale computing and data handling are explained in terms of the complexity of individual collisions and the need for high statistics to study quantum mechanical processes. The prevalence of university-dominated collaborations adds a requirement for high-performance wide-area networks.

The data handling and computational needs of the different types of large experiment, now running or under construction, are evaluated. Software for experimental high-energy physics is reviewed briefly with particular attention to the success of packages written within the discipline. It is argued that workstations and graphics are important in ensuring that analysis codes are correct, and the worldwide networks which support the involvement of remote physicists are described. Computing and data handling are reviewed showing how workstations and RISC processors are rising in importance but have not supplanted traditional mainframe processing. Examples of computing systems constructed within high-energy physics are examined and evaluated.

This review was received in April 1992.

**Contents**

	Page
1. Introduction	1387
1.1. Making measurements in the quantum world of high-energy physics	1387
1.2. Smearing by physics and by detectors: the need for simulation	1388
1.3. Outline of physics analysis strategy	1389
1.4. Outline of the analysis environment for experimental HEP	1390
2. Data volumes and computational needs of HEP	1390
2.1. Data volumes produced by typical detectors (per event)	1390
2.2. Computational needs per event	1391
2.3. Data volumes and analysis strategy for current and future experiments	1392
2.4. Future electron-positron colliders	1395
3. Software for experimental HEP	1395
3.1. Clarity versus efficiency	1396
3.2. The dominance of FORTRAN	1396
3.3. HEP software tools	1396
3.4. How to access terabytes of data?	1399
3.5. Operating systems	1400
4. Workstations and graphics	1401
4.1. What is a workstation?	1401
4.2. Software development on workstations	1402
4.3. 3D graphics: importance and limitations	1402
4.4. Graphics standards: importance and limitations	1404
5. Wide-area networking	1404
5.1. HEP network requirements	1404
5.2. Network performance and cost	1405
5.3. Existing networks serving experimental HEP	1405
5.4. The future of HEP networking	1407
6. Computation and data handling	1408
6.1. The historical role of mainframes	1408
6.2. The RISC revolution	1408
6.3. Local area networks and links	1409
6.4. Data-handling technology	1409
6.5. A model for HEP computing	1410
6.6. Vector and parallel computing	1412
6.7. Software support for parallel computing	1414
6.8. Examples of HEP computing systems	1414
7. Conclusions	1418
Acknowledgments	1419
References	1419

## 1. Introduction

Experimental high-energy physics has a seemingly insatiable demand for data handling, networking and computing. This demand stems both from the fundamental nature of quantum physics, and from the human needs of scientific collaboration on a large scale. This review will examine how problems and constraints arising both from physics and collaboration lead to requirements for data handling, networking and computing. Typical systems constructed to meet these requirements will be described and an attempt will be made to outline the likely evolution of computing for high-energy physics over the next decade.

This review treats only off-line computing which, by definition, takes place after data have been recorded on mass storage. Computing and electronics for data acquisition in high-energy physics is a challenging subject worthy of its own review article. Most quantitative examples will be further restricted to the handful of large experiments which occupy about half of the world's high-energy physicists. This restriction does not distort the picture of the computing activities and computing needs evaluated per physicist.

In describing not just what is done, but why it is done, it will frequently be necessary to present general information about trends in computing which affect high-energy physics.

### 1.1. Making measurements in the quantum world of high-energy physics

High-energy physics (HEP) is one of the fundamental frontiers of knowledge; together with cosmology, HEP explores the limits of our knowledge of the physical nature of our universe. Experimental HEP is, in many ways, experimental cosmology, since the HEP experiments we can perform on Earth appear to re-create the conditions of a universe only picoseconds old. Measurements of physics on the microscopic scale are at least as effective in constraining and guiding cosmology as observations of the macroscopic state of the visible universe.

In practical terms, the observation of the cosmos and the observation of the smallest known objects pose very different problems. The world of elementary particles is a quantum world in which observation creates massive perturbations and in which the results of observations are not deterministic. As an obvious example of the perturbations, the high-energy physicist's normal *modus operandi* is to study something small, for example a proton, by blasting it apart with a projectile carrying kinetic energy equivalent to many times the rest mass of the target. Only at destructively high momenta do probes, such as an electron, have de Broglie wavelengths giving the high spatial and temporal resolution needed to study proton structure.

Probes with destructively high momenta imply high centre-of-mass energy collisions and the production of many high-energy, often penetrating particles. The measurement of these particles requires physically large and highly granular detectors.

The need for massive data handling and computation arises not just from the complexity of individual high-energy particle collisions, but also from the non-deterministic nature of individual collisions. Even if we understood the underlying physics perfectly, we could only predict the probabilities of the various final-state configurations (particle types, momenta and directions) which can arise in, for example, electron-positron collisions at 91.180 GeV centre-of-mass energy. We believe it to be quite impossible to predict that a particular configuration will result from a particular collision. Since,

in reality, we do not understand physics perfectly, we must measure the probabilities experimentally in order to test our theories quantitatively. Measurement of probability implies the accumulation of statistics and thus the physicist's appetite for 'events', or measurements of the results of individual collisions, is almost insatiable. For example, to measure with a 1% statistical error the cross section (i.e. probability) for the process  $e^+e^- \rightarrow Z^0 \rightarrow \mu^+\mu^-$ , requires the accumulation of 10 000 events of this type. To measure the angular distribution of the muons to 1% using ten bins requires about 100 000 events. These  $Z^0 \rightarrow \mu^+\mu^-$  events would be collected along with about 3 000 000 other events, since the dimuon final state occurs in about 3% of  $Z^0$  decays

### 1.2. Smearing by physics and by detectors: the need for simulation

Physics and engineering conspire to obscure our view of what happens in high-energy interactions. We have believed in quarks as fundamental particles rather than as a mathematical convenience since about 1975. However, as illustrated in figure 1, quarks have never been observed directly. We observe only the results of so-called quark fragmentation, a process which smears energies and directions and often leaves little trace of the original quark quantum numbers.

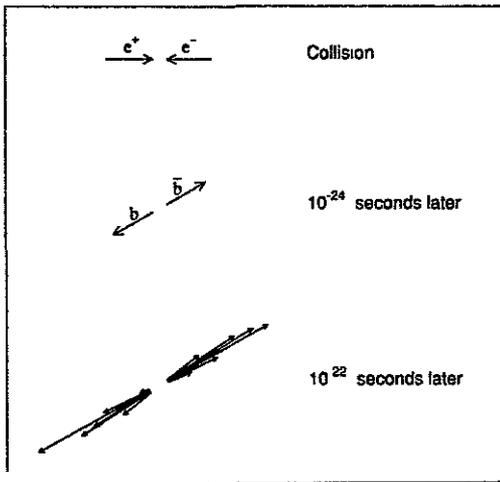


Figure 1. An example of how low-energy interactions obscure high-energy interactions:  $b$  quarks produced in  $e^+e^-$  collisions fragment almost immediately into jets containing  $\pi$ ,  $K$  and  $B$  mesons.

Quark fragmentation is the clearest example of smearing by physics itself. Engineering degrades our measurements by requiring that we leave room for cables, cooling fluids and mechanical supports. Financial limitations are even better at creating temporary or permanent dead regions in our apparatus. Resolution is always finite and sometimes not even money helps. For example, measurement of individual hadrons (pions, kaons, protons, etc) by observing the total energy which they deposit is limited by statistical fluctuations in their interaction with matter to a precision in the region of  $50\%/\sqrt{\text{energy GeV}}$ .

To make precise measurements in spite of smearing, event-by-event Monte Carlo simulation is needed so that we can determine the relationships between the quan-

ties we can measure and the almost hidden fundamental physics which we want to understand.

### 1.3. Outline of physics analysis strategy

Figure 2 shows the main components of physics analysis. The Monte Carlo simulation should be designed to reproduce the real data closely, even to the extent of imitating the known defects of the apparatus. Reconstruction is needed to derive from the readout of hundreds of thousands of detector elements the key quantities which characterize each collision. The final stage of physics analysis involves comparison of the distributions and correlations of these key quantities for both real and simulated events. To search for possible new physics it is necessary to compare real data with simulations including and omitting the hypothetical new effects.

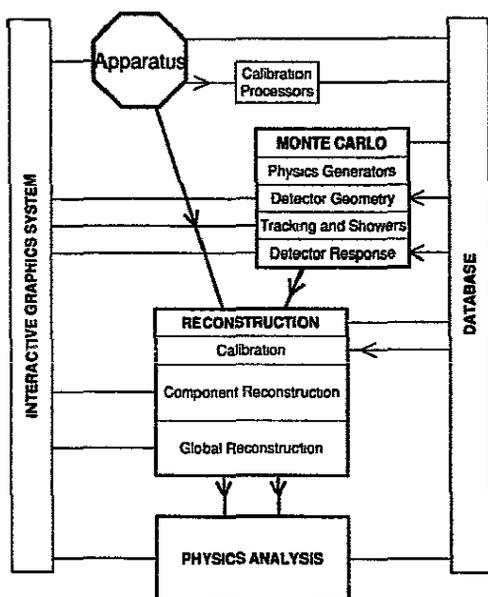


Figure 2. A summary of the strategy for the analysis of data from a high-energy physics experiment. All the vertical data flows involve storage of the data, usually on some form of magnetic tape

Interactive graphics and a database system are the two utilities that allow precise, high-statistics studies to be undertaken. Some form of database is needed to record the varying calibration and status of each of the 100 000 or more detector elements, and to make this information available automatically during analysis. Programs performing Monte Carlo simulation and reconstruction usually comprise hundreds of thousands of lines of newly written code in addition to more standard components. To ensure that the inevitable residual program bugs are not damaging to the results, interactive graphics are used extensively to probe the workings of the code and ensure that all intermediate steps are working correctly. The same tools facilitate a rapid diagnosis of obvious or even subtle problems in the detector itself.

#### *1.4. Outline of the analysis environment for experimental HEP*

Experimental HEP is often called 'Big Science'. Collaborations may comprise 500 physicists who do an experiment involving thousands of tons of apparatus, costing tens or hundreds of millions of dollars, and taking one decade to construct and another to exploit. Perhaps surprisingly, this 'Big Science' is still university dominated. The author's own collaboration, the L3 experiment at CERN, provides a good or even extreme example of this. L3's nearly 500 physicists come from about 42 universities and institutes in 14 countries. The host laboratory provides the LEP accelerator and some very valuable engineering support, but almost no manpower for doing the physics.

In the short term, this geographical and organizational fragmentation of a collaboration is rarely the most efficient way to perform any particular task. Taking a longer view it is valuable or even vital to keep the intellectual leadership of HEP in the universities and thus in contact with a wide range of science, scholarship and teaching. To overcome the attraction of the experimental site by purely administrative pressure alone (e.g. 'no salary unless you stay here and teach') is a sterile approach and pressures of this nature have to be coupled with a determined effort to make access to the data, and thus participation in the experiment, possible from universities.

## **2. Data volumes and computational needs of HEP**

HEP computing differs from many types of 'scientific computing' in the importance attached to data handling in addition to pure computation. As pure computation has become more and more available over the last ten years, high-energy physicists have become increasingly aware of the importance of data handling, which now dominates the cost of HEP computing. We will examine the data-handling and computational needs by looking first at what is needed to record, analyse or simulate one event, since the per-event needs of many current and future experiments are rather similar.

### *2.1. Data volumes produced by typical detectors (per event)*

A large HEP detector capable of detailed measurements of collisions in the range 100–1000 GeV centre-of-mass energy has of the order of 200 000 sensitive elements which are read out whenever a trigger system† detects a wanted event. A single sensitive element may produce between one and one hundred bytes of information. Thus the uncompressed readout from a large detector amounts to over 1 Mbyte/event. However, detectors have large numbers of sensitive elements so that they can readily distinguish one energy deposit from its neighbours. In consequence, the majority of elements are not 'hit' by any particle in a particular collision and the readout from an element which shows a response consistent with zero is normally discarded. After this compression the data volume is reduced to 100–200 kbyte/event.

Future detectors for higher energies, such as 16–40 TeV, will have more sensitive elements so that they can resolve the more complex events. Event sizes of about 1 Mbyte are expected for such detectors.

The number of bytes per sensitive element is not a constant of nature. Fifteen years ago, most sensitive elements produced exactly one bit of data (1 = 'something

† A system of high-speed analogue and/or digital electronics which uses a subset of the signals from the detector to make an accept/reject decision.

hit me'). Today, many elements are read out by 'flash ADCs' which record the output of the element as a function of time—for example, 100 measurements of 8-bit precision. This change has been caused by a continual optimization of the relative spending on hardware, readout electronics and computers, taking into account the relative stability of the hardware costs as compared with a steady fall in the cost of electronics and computing. There is no sign that this evolution has stopped.

## 2.2. Computational needs per event

The computationally demanding elements of the analysis strategy already shown in figure 2 are the reconstruction and the simulation. HEP programs are not naturally vectorizable, rather they present a mixture of logic, integer arithmetic, 32-bit and 64-bit floating-point arithmetic in approximately that order of importance. Computational needs will be described in terms of HEP-MIPS†, a unit which is approximately defined, by giving a few examples, in table 1.

Table 1. HEP-MIPS ratings examples of the relative performance of computers for typical HEP programs.

Computer	HEP-MIPS rating
IBM 3090-600E (1 processor)	25
Cray XMP-48 (1 processor)	30
Apollo DN10000 (1 processor)	12-15
HP 9000/720	24-35
IBM RS6000-320	10-13
VAX 11/780 (with floating-point accelerator)	1

The first step in the physics analysis of a few or a few million events is the reconstruction of each event. In a typical event, a few tens of particles emerging from the primary collision produce tens of thousands of energy deposits recorded by individual detector elements. Ideally, reconstruction would derive the exact energy, direction, and particle type of each of these particles. When this is impossible, for example when several particle paths are so close as to remain unresolved, then reconstruction should be able to give a good estimate of the direction and magnitude of the energy flow of this unresolved 'jet'.

A reconstruction program uses pattern-recognition algorithms to identify sets of raw energy deposits which were probably due to a single particle or jet. The logical relations between jets, particles, track segments, energy clusters and the raw energy deposits are recorded in a data structure. Thus later analysis has access both to the reconstructed jets and particles and to their components and subcomponents as determined (or guessed) by the reconstruction program.

Reconstruction of a typical event from a general-purpose detector for 100–200 GeV collisions requires 50–150 HEP-MIPS s. Reconstruction is now always done (at most) a few hours after data acquisition since it provides a detailed check on the quality of the data. Inevitably, later improvements to calibrations or software make it necessary to repeat all or part of the reconstruction. Inclusion of the computing

† HEP-MIPS units are similar to the SSCUPS units used by the Superconducting SuperCollider Laboratory, and to VUPS (VAX units of performance) as frequently used in HEP computing. HEP-MIPS may differ by more than a factor of two from manufacturers' MIPS ratings.

needs of the later stages of physics analysis can bring the total amount of computation performed per event to as much as ten times that of the initial reconstruction.

The computational load of simulation has changed greatly in the last 15 years for reasons very similar to those which have increased the number of bytes per sensitive element. Fifteen years ago, simulation of the response of a detector to particles would normally have used a simple description of the 'live' and 'dead' zones, plus parametrizations of the most probable response of the live parts. Passing all the simulated events through the reconstruction was recognized to be a good idea, but was considered too expensive, so the simulation program would produce a summary of what the reconstruction program might have found.

Today, the needs of a generation of high-precision experiments, together with the revolutionary decrease in the cost of pure computation, have increased the CPU time spent in Monte Carlo simulation by three or four orders of magnitude. Per simulated 100 GeV event, 2000–5000 HEP-MIPS s are spent following particle paths through the millions of insensitive and sensitive detector elements, and to simulate in great detail the processes by which most particles interact and re-interact to deposit all their energy as a 'shower' of daughter particles. The output is a simulation of the detector readout and it must be passed through the reconstruction program before it can be compared with real data.

The CPU time required to simulate events is dominated by detailed, event-by-event shower simulation. If precision is maintained, the time required for shower simulation is closely proportional to the total energy deposited in the detector and thus at 40 TeV, it can take hundreds of HEP-MIPS hours to simulate one event.

### 2.3. Data volumes and analysis strategy for current and future experiments

Table 2 summarizes the data volumes and CPU needs for a range of current and future experiments which will be outlined. Event rates and running times are used to calculate absolute data volumes and CPU needs from the per-event needs already described.

Running times per year for major experiments are all broadly similar. Physics is normally scheduled for about half of each year. During the scheduled time accelerators run 24 hours/day, but with (from the physicists viewpoint) an efficiency which rarely exceeds 50% due to the need to stop and re-fill with particles and due to component failures. Thus accelerator-based experiments are able to take data for about 2000 hours/year.

Table 2 also contains, implicitly, estimates of the factor by which the reconstruction time per event must be multiplied to take account of subsequent analysis steps and re-reconstruction. This factor is normally less than ten, and can be quite small for some proton-proton experiments where many events are discarded after the first pass of the reconstruction program.

**2.3.1. The LEP experiments:  $e^+e^-$  collisions at about 100 GeV.** The ALEPH, DELPHI, L3 and OPAL detectors record collisions produced by the LEP electron-positron ring at CERN, Geneva. Electrons and positrons are simple, perhaps fundamental, 'pointlike' particles with no known substructure. Their collisions are difficult to achieve but relatively easy to interpret due to the well understood initial state. When it is possible to observe large numbers of electron-positron collisions, the full statistics of the sample can be used to make precision measurements of fundamental physical parameters or, equivalently, precision tests of physical models.

Table 2. A summary of data volumes and CPU needs for current and future large experiments. The figures are per experiment during the years of operation.

	Current experiments			Future experiments	
	$e^+e^-$	$e^-p$	pp	pp	$e^+e^-$
Centre-of-mass energy (GeV)	100	300	2000	16 000–40 000	500–2000
Raw rate of interesting events (Hz)	$\lesssim 1$	1–1000	$1-10^6$	$1-10^8$	$\lesssim 0.001$
Event rate written to 'tape' (Hz)	$\sim 1$	$\sim 1$	$\sim 1$	10–10 000	$\sim 1$
Event size (kbyte)	100–200	$\sim 100$	$\sim 100$	1000–10 000	1000–10 000
Reconstruction: time per event (HEP-MIPS s)	50–150	$\sim 100$	$\sim 200$	500–5000	500–5000
Detailed simulation: time per event (HEP-MIPS s)	2000–5000	2000–5000	$\sim 5000$	20 000–200 000	20 000–100 000
Annual data volume (Tbyte)	1–10	1–10	$\sim 10$	$\sim 1000$	$\sim 10$
Reconstruction and analysis: CPU needs (HEP-MIPS)	50–150	30–100	30–100	$\sim 100 000$	$\sim 200$
Simulation: CPU needs (HEP-MIPS)	200–2000	100–1000	50–500	$\sim 100 000$	$\sim 200$

At the LEP accelerator up to one event per second occurs within each experiment when the machine energy is tuned close to the mass of the intermediate vector boson  $Z^0$ . At other energies the rates are much smaller and when, as is planned, the LEP energy is increased towards 200 GeV, rates of one wanted event per hour will be considered good, although unwanted backgrounds will keep the rate of data acquisition close to one per second. Each LEP experiment thus acquires about  $10^{12}$  byte (1 Tbyte) of raw data per year, and when running (as now) at the  $Z^0$  mass, almost all of these data consist of interesting collisions to be reconstructed and made available to hundreds of physicists for analysis. Expressed in terms of physical media, hundreds of physicists would like to have almost random access to tens of thousands of (200 Mbyte) tape cartridges.

The opportunities for high-precision physics can only be exploited by comparing real data with high-statistics samples of simulated data. There is no justification (at least for data samples of a few million events or less) for failing to generate several times as many simulated events as data.

*2.3.2. Other current collider experiments.* The CDF and D0 proton-proton experiments at the 2 TeV Tevatron at the Fermi National Accelerator Laboratory near Chicago, and the H1 and ZEUS electron-proton experiments at the HERA accelerator at DESY, Hamburg are all comparable in complexity to the LEP experiments.

Protons are complex objects and the proton-proton collision cross section is a million times greater than the (inelastic) electron-positron cross section at the  $Z^0$  resonance. However, proton-proton collisions capable of uncovering new physics are rather rare. These collisions occur when a pointlike constituent of one proton, which happens to be carrying a large fraction of the proton's momentum, hits a similarly endowed constituent of the other proton. Analysis of proton-proton data is like a search for needles in a haystack. It is obviously impossible even to record the complete sample of collisions (millions of Tbytes), and both the on-line and off-line analysis consists of a series of filters which aim to maximize the value of the retained data. These filters implement necessarily prejudiced decisions about which events are interesting and the success of an experiment depends very strongly on the skill and intuition embodied in their design.

With this in mind, it is understandable that the amount of data recorded in a proton-proton experiment is usually the maximum which can be handled with a tolerable cost and inconvenience. A moment's reflection on the tens of thousands of tape cartridges produced by the LEP experiments will convince most people that this represents about as much inconvenience as it is wise to tolerate. Thus the Tevatron experiments, with similar event sizes to those at LEP, have chosen tape-writing rates close to one event per second.

Since the two extremes, electron-positron collisions and proton-proton collisions, produce very similar quantities of data for off-line analysis, it is not surprising that the electron-proton experiments now starting at the HERA accelerator expect to acquire data at about the same rate. However, only LEP experiments have the luxury and duty to acquire all the collisions and examine them at leisure. Although the electron-proton cross section is much smaller than that for proton-proton collisions, HERA experiments must still reject the less interesting events in real time to arrive at a tolerable tape-writing rate.

The very different collision rates of electron-positron, electron-proton, and proton-proton machines have been constrained by data storage technology to have

comparable rates of data acquisition, and hence comparable computing needs for the treatment of real data. Different considerations hold for Monte Carlo simulation. The high-precision measurements that are possible using electron-positron collisions justify the generation, using very detailed simulation, of a larger number of simulated events than real data events. Our poorer understanding of the majority of proton-proton collisions implies that there is normally no point in attempting a painstaking simulation of the full data sample. Physics analysis concentrates more on smaller selected event samples and the computational needs for simulation are relatively smaller than for electron-positron experiments.

**2.3.3. Future proton-proton colliders.** The SSC, a proton-proton collider with 20 TeV beams, is now under construction in Texas and will be operational early in the next century. The LHC, a slightly lower energy machine using much of CERN's LEP infrastructure, may be built on an even shorter timescale. When searching for new physics in proton-proton collisions, the ratio of needles to hay improves as the energy increases. The proponents of the LHC hope to make it a competitive machine by achieving a very high collision rate—tens of collisions (all mixed together) every 15 ns. This high rate (corresponding to a luminosity of  $10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ ) will pose formidable problems for off-line and especially for on-line data analysis.

Numerical estimates for SSC/LHC computing are shown in table 2. The enormous uncertainty in the amount of data to be recorded reflects uncertainty about the evolution of mass storage technology. Using only the mature technology of today it would be difficult to store data from more than  $10^{-9}$  of the collisions which occur. If in ten years from now, for the same real cost,  $10^4$  times as much data could be stored, the event filters would certainly be re-tuned or re-designed to accept at least  $10^{-6}$  of the collisions. The wider range of events available for analysis would bring significant benefits to the physics.

#### **2.4. Future electron-positron colliders**

The advantages of intelligible measurements leading straight to fundamental physics make it almost certain that an electron-positron collider in the 1000 GeV range will be built some day. However the technical problems are daunting, and the first (perhaps the only) such machine will be constructed well after the SSC. The rate of interesting events at such a machine will be very low unless we are able to make unexpected progress in accelerator design. The need to record and understand many background events would lead to a computing load which is 'respectable' by today's standard, but which may be quite tractable by a desktop computer of the twenty-first century. While this should not be interpreted as a prophesy that HEP computing will cease to be a challenge by 2010 it should be clear that, on the timescale of decades, there could be large fluctuations in the relative importance of computing in HEP.

### **3. Software for experimental HEP**

There are normally very few (or even no) professional computer programmers in an HEP collaboration, yet a large collaboration will write hundreds of thousands of lines of code for its 'software base', plus perhaps millions of lines for individual use. Most high-energy physicists are programmers and all of them are perfectly aware that neither their own code, nor that in the software base, can possibly be bug-free.

No attempt will be made to give a thorough review of informal and formal software engineering methods in HEP. At a detailed level this is always a controversial subject and a serious treatment would require a complete review article. Below we will examine just a few of the key points. For more information the reader is referred to the conference proceedings Lyon-1990 and Erice-1990.

### 3.1. Clarity versus efficiency

In the not too distant past some physicists may have measured their worth by their ability to cram as much computation as possible into the minimum number of lines of code and the minimum amount of precious computer memory. Efficiency is still important; examples can easily be given of programs that have been written to run correctly but 10000 times slower than they should. However, squeezing out the last 30% of performance in exchange for a serious degradation in intelligibility is no longer considered an achievement. Most collaborations try to (self-)impose a formal or informal straitjacket on each programmer such that an individual's style or even brilliance is suppressed and new graduate students, or even old professors, can understand the resultant code in a reasonable time.

### 3.2. The dominance of FORTRAN

High-energy physicists have been using FORTRAN since it was invented, and since almost every physicist is a programmer, it is easy to see how a decision to change to another language is almost impossible to implement. Nevertheless, there is continual debate (e.g. White 1989), and every laboratory shows verbal or physical manifestations of the 'there must be a better language than FORTRAN' idea. In self-contained areas, C is increasingly popular, especially as it appeals to the average physicist's desire to have total control and be quite capable of doing dangerous things if he so chooses.

FORTRAN is very appropriate for scientific computing dominated by calculation, but as we noted in section 2.2, calculation does not dominate HEP computing. HEP has managed to live with FORTRAN by writing its own tools to combat the language's deficiencies, particularly in the areas of data-structure management and machine-independent input-output. The existence of these tools is a further brake on a move to another language and even puts in question the extent to which the new features of FORTRAN90 will be exploited in the near future.

### 3.3. HEP software tools

HEP software tools fall into three categories:

(i) *Tools written to overcome limitations of FORTRAN.* Most notable is the ZEBRA package (Brun and Zoll 1987) which provides sophisticated data-structure management and machine-independent sequential and random I/O. The facilities offered by ZEBRA are arguably superior to those offered by any language which might be considered as an alternative to FORTRAN. ZEBRA is both loved and hated, occasionally by the same people. The principal cause of the dislike is probably that ZEBRA is not an integrated part of FORTRAN and its use must be understood and debugged with only minimal help from the FORTRAN manual. Perhaps as a relic from the days when efficiency was vital, the ZEBRA approach to storage and retrieval of data within a structure uses FORTRAN assignment statements rather than subroutines or functions. To FORTRAN, the whole ZEBRA data structure is just a single gigantic vector, any part of which can be directly accessed, or corrupted, by the application.

Other tools in this category are HYDRA (Zoll 1989†) and ZBOOK (Brun *et al* 1984) (precursors of ZEBRA), and BOS (Blobel 1988). ZEBRA is used by over half of the world's large HEP experiments and BOS by a large fraction of the remainder.

(ii) *Tools providing machine-independent functions which are (or were) commercially available only in machine-dependent implementations.* The prime example is the PATCHY (Klein and Zoll 1980) code-management system. PATCHY is very, very old and still has a strong punched-card flavour. For more than 20 years it has provided code management and exchange on every type of computer high-energy physicists have ever used. Recently the creation of CMZ‡, (Brun *et al* 1989a) a PATCHY compatible and portable code-management system able to take full advantage of a workstation environment has, if anything, prolonged the life expectancy of PATCHY itself.

The second example in this category illustrates a number of interesting points. HEPDB is a database management system aimed at the calibration and bookkeeping needs of an experiment. Such databases range in size from 10 Mbyte to 1 Gbyte, many orders of magnitude smaller than the event 'database'. HEPDB does not really exist yet, but it will be a combination of the concepts and code from two relatively mature existing systems, DBL3 (Adeva *et al* 1991a, b) and OPCAL (Cranfield *et al* 1991), each of which was written to meet the needs of a single LEP experiment. HEPDB is built on top of the ZEBRA random access features, giving it effortless machine-independence and is optimized to support access via FORTRAN. The use of HEPDB does not imply that the application must also use ZEBRA to manage data, but it cannot be denied that this is the most natural approach. HEPDB meets the HEP need to have an experiment's database present all over the world through a system which collects all updates centrally and then broadcasts them to remote 'slave databases'.

But, one might argue, 'machine-independent database systems have been on the market for about ten years; why waste time writing your own?'. The answer to this question has two components:

- (i) HEPDB and its antecedents have some features which are not present, or are not as efficiently implemented, in any commercial system;
- (ii) installing HEPDB on a new computer system is technically, administratively and financially trivial. Commercial systems normally require dedicated manpower, and involve the administrative and financial cost of obtaining a licence. It should also be noted that installation of database systems in many countries is regulated (and thus often impeded) by COCOM§.

The evaluation of commercial systems which resulted in the decision to write the DBL3 system took place in 1984. Since that time the purely technical arguments against commercial systems have weakened, but many of the administrative and financial obstacles remain.

The third example in this category is the CERN KERNLIB library. This library contains about 600 mathematical and data-handling entries, has been implemented on almost every type of computer, and is a prerequisite for almost all CERN packages. There are some very respectable commercial alternatives for parts of KERNLIB, such as the NAG Library||, but, as indicated above, in HEP there are great disadvantages in the choice of a commercial package for collaboration-wide use.

† In references to software documentation the date of the most recent version is always quoted. The original software package may be much older.

‡ CMZ is a product of CodeME SARI, St Genis-Pouilly, France.

§ Committee for Multilateral Export Controls.

|| A product of the Numerical Algorithms Group, Oxford, England.

Until now 'machine-independent' has meant running under UNIX (all flavours), VAX/VMS, IBM-VM/CMS, IBM-MVS, Apollo-AEGIS and many other operating systems. If UNIX really does† come to dominate HEP, machine independence will be much less of a problem in the future.

(iii) *HEP specific tools.* This large category includes programs or utilities which may have been written specifically to meet needs recognized as common to many experiments, or may have been written initially by and for a single experiment.

Many components of event simulation are common to a large number of experiments. Event generators are packages which simulate known or hypothesized particle production in individual  $e^+e^-$ ,  $e^-p$ ,  $pp$  or  $p\bar{p}$  collisions. Where possible, generators use perturbative calculations based on the summed effects of many Feynman diagrams. Where, as in quark fragmentation, the problem is intrinsically non-perturbative, phenomenological models are used and adjusted to agree with observation. Examples of widely used event generators are:

(a) KORALZ (Jadach *et al* 1989) generates  $e^+e^- \rightarrow Z^0 \rightarrow$  fermion pair. The program KORALB (Jadach and Was 1985) generates  $e^+e^- \rightarrow$  fermion pairs at lower energies.

(b) JETSET (Sjöstrand 1982, Sjöstrand and Bengtsson 1987) which implements the 'Lund' colour string fragmentation model for quarks produced in  $e^+e^-$  collisions.

(c) HERWIG (Marchesini and Webber 1988, Knowles 1988, Catani *et al* 1991, Abbiendi and Stanco 1991, Marchesini *et al* 1992) which implements a cluster fragmentation model and also includes a simulation of  $e^-p$  collisions at HERA energies.

(d) PYTHIA (Bengtsson 1984, Bengtsson and Sjöstrand 1987, Sjöstrand and van Zijl 1987) which implements the colour string model for proton-proton collisions at LHC/SSC energies.

(e) ISAJET (Paige and Protopopescu 1986) which simulates proton-proton collisions at LHC/SSC energies.

The GEANT package (Brun *et al* 1989b, Brun and Carminati 1991) is now a component of most Monte Carlo simulation programs for current and future detectors. Application software uses GEANT subroutine calls to create a data structure containing a detailed detector geometry. GEANT tracks particles through the detector allowing them to interact according to appropriate cross sections with the sensitive or insensitive components of the detector. GEANT passes a list of energy deposits to the application which can calculate the digital readout from the sensitive elements. Detectors with millions of components are relatively easy to describe assuming a typical level of symmetry.

In the last ten years the EGS program (Nelson *et al* 1985) and the GHEISHA program (Fesefeldt 1985) have been recognized as the best available simulations of the interactions of electromagnetic and hadronic particles with matter. EGS acted as the standard against which the electromagnetic simulation in GEANT was checked. GHEISHA, although still available as a separate program, has now been incorporated into GEANT as the GEANTH package.

The FATMEN dataset catalogue (Shiers and Goossens 1991, Shiers 1991) is a newer creation designed to address the problem of managing and accessing hundreds of thousands of datasets distributed (and perhaps duplicated) over many computers. The core of FATMEN is a database of datasets. FATMEN as perceived by users is both this database, and the (often pre-existing) utilities for dataset retrieval and copying

† See section 3.5.

which can be invoked by FATMEN to bring data from, for example, a tape drive on a remote computer, to a user's program.

PAW (Brun *et al* 1989c, d) is an ensemble of packages for interactive data analysis including the generation and display of histograms and more complex plots. In its current version, PAW requires that the physicist writes a program to extract from the complex data structure describing each event a relatively small number of variables per event (an 'Ntuple') which can then be used interactively as the basis for PAW selections and displays. PAW is widely used but its authors and its users would like it to be able to read and manipulate data structures more complex than the fixed-format Ntuples.

It should be clearly understood that successful software packages, although occasionally the result of a coherent management of effort at a single laboratory, have never been products of HEP-wide planning. Many of the most successful packages were created without the knowledge (or even against the better judgment) of 'the management', and most physicists are not in favour of rigid, long-term, worldwide planning. Nevertheless, the advantages of cooperation and information exchange among the people who write the code have become clear, and a relatively informal 'HEPLIB' collaboration now links the major European and USA laboratories.

Most of the above examples of important software packages are products of CERN or the CERN user community. This choice of examples is natural since these are the packages that are well known to the author and can be described without distortion. However, it must be admitted that HEP appears to be passing through a period where CERN packages may be more widely used in the world than packages from the sum of all other laboratories. This is partly due to CERN's possession of an accelerator and experiments which are both demanding and very productive of physics results, but it is also due to the presence at CERN of a few highly motivated people having excellent contact with their community of users.

### 3.4. How to access terabytes of data?

Now, and probably for the next 20 years, the big challenge in HEP computing is how to give hundreds of physicists access to far more data than can be stored on random access devices at tolerable cost. The partial solutions to the problem rely on the fact that all data are not equal. Some events are much more in demand for careful study than others, and within each event, there are some raw or reconstructed quantities which are referenced frequently, and other quantities which are rarely used. If an experiment's data and physical resources (say 100 Gbyte of disk space, 1 Tbyte of robot-mounted tapes and 10 Tbyte of manually mounted tapes) were put in the hands of a sufficiently sophisticated (psychic?) database-management system, then the average access time to the whole 10 Tbyte could be kept very close to that for disk-resident data. Today's general-purpose database technology has problems with databases bigger than a few Gbytes; although HEP remains in contact with the commercial database industry, both sides agree that our current and our future data-management problems are rather too large for treatment with general-purpose software.

In partially solving the data-management problem with our own software, we rely heavily on somewhat inflexible decisions about what will be interesting and what less interesting. In its most extreme form, this can mean selection of 1% of the data to keep on disk, leaving the other 99% inaccessible to all but the most determined. A more flexible approach is outlined in figure 3 which shows how the L3 experiment

splits up (SPLT) its data into about 27 non-exclusive data streams, some containing full events and some containing only the more commonly used parts of the reconstruction data structure. These data sets are concatenated (CCAT) and stored in  $\sim 200$  Mbyte chunks whose retrieval from tape and storage on disk is managed by simple algorithms. The data handling shown in figure 3 proceeds without any attempt to produce 'cleaned-up', ordered datasets since to do this would mean keeping hundreds of Gbytes of data on disk for weeks or months. The status (good, superseded, or bad) of the batch jobs which processed every event on a CCAT output dataset can be obtained from the L3 database system.

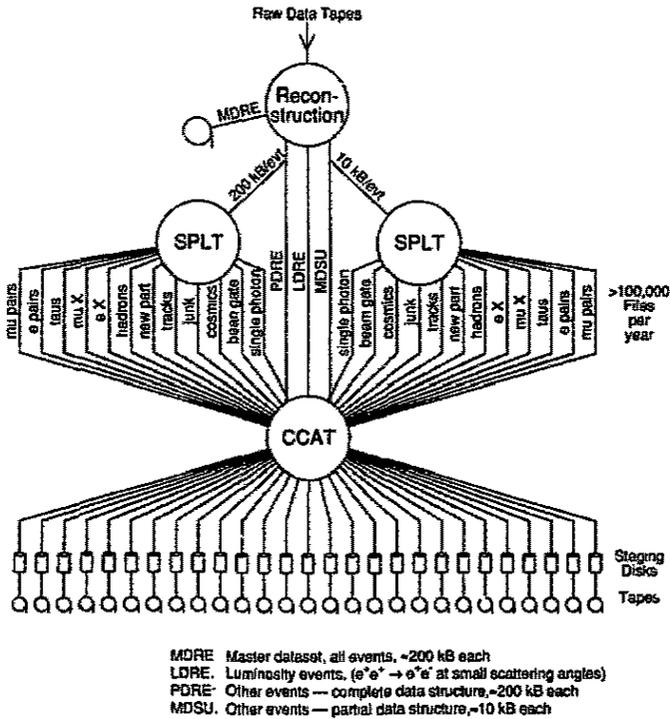


Figure 3. Management of reconstructed event data for the L3 experiment. The program SPLT splits up the data into 13 non-exclusive event streams. The program CCAT concatenates the SPLT output files and stores them on cartridge tapes and on temporary 'staging' disks for access by workstations.

Whatever the details of the data selection employed by an experiment to ensure that the needed data are normally available, all systems need a data catalogue to give physicists access to data in a location-and device-independent way, together with networks to move data quickly from the point of long-term storage (often a mainframe) to the point of use (usually a workstation).

### 3.5. Operating systems

The joke that 'the average physicist can remember 1.1 operating systems' is now very old, but many a physicist, as he strikes the wrong key and loses ten minutes' work, is

reminded of it daily. The average physicist in the L3 collaboration has to be familiar with VAX/VMS used on the on-line data-acquisition cluster, with IBM VM/CMS used on L3's and CERN's IBM mainframes, and with AEGIS and/or UNIX used on the collaboration's workstations. The historical choice of all of these systems was either inevitable or at least defensible, but it must be hoped that the future will be a little simpler. Some operating systems, such as VM/CMS, were inadequate in their original form, but have been modified and supplemented by HEP laboratories to make them powerful and reasonably user friendly.

Until the surge in workstation use in about 1987, UNIX was almost unknown to high-energy physicists. Even now, most physicists use UNIX as little as possible since it is considered unfriendly and normally lacks features such as tape handling and sophisticated batch scheduling which have long been considered essential. However, nearly all the HEP-MIPS are now available in workstations running some flavour of UNIX, so we are reasonably certain that UNIX is here to stay. Can we make life simpler by getting rid of the other operating systems? The problem machines are principally IBM mainframes performing vital data-intensive tasks. Trials of early versions of AIX, the UNIX-compatible operating system for these mainframes, have shown that it will be some time before AIX for mainframes has the necessary features and stability. Nevertheless, physicists have become resigned to or enthusiastic about an eventual complete conversion to UNIX. The HEP-IX collaboration, between UNIX support groups in the various HEP laboratories, aims to achieve a coordinated effort on the utilities and enhancements which are considered essential.

#### 4. Workstations and graphics

In this section we will give a definition of a workstation and describe its impact on software development and graphics. The role of workstations in organized computation and data handling will be addressed in section 6.5.

##### 4.1. *What is a workstation?*

In HEP a workstation is a computer which has sufficient memory and CPU power to support the development and execution of HEP software and is optimized for use by one person. Up to now this definition has excluded PC and Macintosh computers which, while widely used for writing letters or drawing diagrams, had insufficient virtual and real memory to run HEP software. In 1987, a workstation which was both affordable and useful had typically a 1280 pixel  $\times$  1024 pixel, 19 in monochrome screen, 2 Mbyte of real memory, and 1 HEP-MIP of CPU power. An average of 100 Mbyte of disk space per workstation was adequate, at least for experiments which had no real data and were only developing software.

Today a 'low end' (\$7000) workstation has the same 1280  $\times$  1024 monochrome screen, 16 Mbyte of memory, 400 Mbyte of disk and 12 HEP-MIPS of CPU power. More powerful machines, still affordable (< \$30 000) by many university groups, bring a 1280  $\times$  1024 colour screen with up to 40 colour planes, 3D graphics at 1 M vector/s, over 1 Gbyte of disk and at least 30 HEP-MIPS of CPU power.

Yesterday's 'low end' machines tend to be used largely as sophisticated terminals, performing editing and some compilation and physics analysis, but initiating all CPU-intensive tasks on a more powerful machine. The value of dedicated X-terminals is not always clear, although one USA laboratory has now issued guidelines stating that

buying one powerful workstation plus a few X-terminals is the best way to provide for the needs of a university group.

#### *4.2. Software development on workstations*

When the preparation of the LEP experiments began in 1982, many physicists were very worried at the prospect of attempting precision measurements with detectors and software which were significantly more complex than anything they had encountered before. Fortunately, workstations and relatively plentiful computing power have transformed HEP computing; the LEP detectors are even more effectively exploited than the less complex detectors of the 1970s.

Workstations have greatly shortened the edit-compile-link-run cycles of software development and provide interactive debugging features vastly superior to anything offered on a simple terminal. Most importantly, workstations can make it easy to exploit graphics to understand what software is doing. Every component of a program can be made to produce results suitable for graphical display, taking advantage of the human brain's much greater ability to see anomalies in pictures rather than in tables of numbers.

The advantages of workstation graphics do not come for just the price of the hardware, it is essential that the application software be designed, almost from the start, to exploit these advantages.

Figure 4 shows a screen generated by software designed in this way. The display is generated by the L3 reconstruction/analysis software. It includes a 3D representation of some of the energy deposits and reconstructed quantities in an event together with the entry points to over 200 menus which can control the display and the reconstruction program, and probe deeply into the workings of the program. As a simple example, the list of energy deposits in a cluster of crystals (the lower right window in the figure) was displayed as a result of selecting the graphical image of the cluster with the workstation's mouse.

#### *4.3. 3D graphics: importance and limitations*

In the 1960s, when bubble chambers were the most widely used detectors, physics results were often obtained by looking at and classifying events. Today physicists take care to point out that no such subjective analysis is involved in obtaining results. Graphics help us to understand what our detector is really doing and what our software is really doing. Rarely, if at all, do event displays help us to understand physics, although histograms and plots presenting a statistical analysis of many events are widely used in physics analysis.

Detectors are undeniably 3D objects, but much of the readout has a projective geometry. For example, many wire chambers provide no information about how far down a wire the energy deposit occurred. The readout from such detectors is most conveniently drawn when projected onto a plane; indeed trying to show it in real 3D space can be very difficult and confusing. In contrast, the goal of a reconstruction program must be to calculate the real position in 3D space of each energy deposit, and a display of the program's final or intermediate results can benefit from 3D graphics. Experience shows that a performance of close to 1M 3D vector/s is adequate to display and manipulate the optimum quantity of information on a screen of 1280 × 1024 resolution. Colour is vital, but we know of no event display application which has made serious use of surface drawing or lighting models.



Up to now, the high cost of displays or workstations capable of 1 M 3D vector/s has restricted their use. A few experiments, such as L3, have 3D graphics available on a large fraction of workstations, but in most experiments 2D representations of the data are used skilfully to convey information to the physicist. The 3D graphics accelerator that used to cost \$20 000 a few years ago is now a standard feature of a \$12 000 workstation. Thus even if the '3D or not 3D' debate continues, it has now been deprived of its financial component.

#### 4.4. Graphics standards: importance and limitations

Standards can be wonderful, but only if they are well supported by the computer industry. In the 1970s, HEP use of computer graphics was plagued by a complete absence of standards and by our failure to impose our own locally invented systems on our friends. In the 1980s the GKS standards started to emerge, but 2D GKS received poor support from industry and 3D GKS was practically ignored. Nevertheless, in Europe, HEP management attempted to promote the use of GKS with some useful results for 2D graphics. The effective absence of a 3D standard helped to discourage wide use of 3D graphics.

One reason for industry's lack of interest in GKS-3D was the large investment that many manufacturers had just made in hardware supporting a hierarchical graphics structure. GKS could not exploit this hardware effectively and, at a time when few machines had graphics performance to spare, was unacceptable to vendors and users alike.

The PHIGS† hierarchical graphics standard now appears to have gained wide acceptance by industry and is perfectly adequate for HEP graphics, even if many incompatibilities exist between different implementations. When allied to the MOTIF‡ industry (as opposed to ISO or ANSI) standard for a user interface, it becomes possible, for the first time, to plan an interactive graphics system that will run almost unchanged on a wide range of hardware.

## 5. Wide-area networking

### 5.1. HEP network requirements

The aim of HEP networking is to remove all the network-related disadvantages of working in Amsterdam or Bristol rather than CERN or DESY. Today, most physicists at CERN can get access to data at speeds of a few tens of kbytes  $s^{-1}$  (to within a factor of ten). The physicists are not happy with these speeds, but the waiting is acceptable because it also applies to their colleagues. As we will see in section 6.3, some local area links and networks providing around 1 Mbyte  $s^{-1}$  to a user are already becoming available. As soon as their use is widespread this higher figure will become the wide-area requirement.

It is relatively cheap to achieve the currently required bandwidth by sending parcels of tapes by air freight. Air freight is perfectly appropriate to support rigidly planned activities, but it is no way to respond to a remote physicist who has just had an idea. Nevertheless, air freight is still forced upon most university groups by

† Programmers Hierarchical Interactive Graphics System (ANSI/ISO standard).

‡ Open Software Foundation.

the cost of networking. The use of national computer centres, to which tapes are systematically sent from the experimental site, can make the air-freight solution less of a disaster.

If deprived, by air freight, of equality with their on-site colleagues, remote physicists still have a need for network access to their experiment's software base and database and to be able to make some interactive use of the computers at the experimental site. A few kbyte  $s^{-1}$  (on demand) is adequate for this, although at these speeds X-Windows, if used at all, must be used with great care.

An average university group with several active graduate students needs at least 64 kbit  $s^{-1}$  to meet the basic needs and at least 256 kbit  $s^{-1}$  (today) to support access to data. A trunk supporting ten universities and a large group at a national laboratory should have at least four times these capacities today and planning should have started for an upgrade of at least one order of magnitude to meet future needs.

### 5.2. Network performance and cost

Lines at up to 2 Mbit  $s^{-1}$  (or 1.544 Mbit  $s^{-1}$  in the USA) are available between most points in Europe and the USA at one or two month's notice. Higher speeds ( $\gtrsim 34$  Mbit  $s^{-1}$ ) can also be obtained with a little more effort. The cost of a 1.544 Mbit  $s^{-1}$  line between Los Angeles and Boston (5000 km) is about \$200k per year. In sad contrast, the cost of a 2 Mbit  $s^{-1}$  line between two randomly chosen points in Europe is likely to be about \$1 million per year.

### 5.3. Existing networks serving experimental HEP

Figure 5 shows the international network lines which terminate at European institutes involved in experimental HEP. Most of these lines are used primarily for experimental HEP, but the 1.544 Mbit  $s^{-1}$  line to Cornell University in the USA is an example of a line funded from non-HEP sources which has been installed to take advantage of CERN's *de facto* status as a European networking hub. The large range of speeds of the lines radiating from CERN reflects the fact that all such lines are entirely funded by the agency responsible for the remote end, and are thus subject to a wide variety of funding and decision-making processes. HEP traffic dominates most of the lines which reach CERN, but within countries, some (such as the UK) use a general-purpose academic network, while others (such as France) use dedicated HEP lines. More details of trends in European HEP networking are given by Fluckiger (1991).

Figure 6 shows the ESNET (Energy Sciences Network) in the USA. The US Department of Energy is the principal (but not the only) funding agency for the USA HEP which shares ESNET with other DOE-funded users. The 1.544 Mbit  $s^{-1}$  backbone of ESNET is adequate for current use and plans exist (even if funding is always uncertain) for an upgrade to 45 Mbit  $s^{-1}$  when this is needed. Some perspectives on USA HEP networking are given by Lidinsky (1991) and as befits a nationally organized network, ESNET has a program plan (Cavallini *et al* 1991).

The 512 kbit  $s^{-1}$  line between CERN and MIT is managed by the L3 collaboration rather than by CERN or the ESNET management since this line was originally funded to meet the needs of US participants in the L3 experiment at CERN. The CERN-MIT line is complemented by dedicated lines (not shown in figure 6) from MIT to several US universities.

Up to this point no mention has been made of protocols. HEP tries to avoid the dogmatic disputes which appear to occupy the time of many networking experts.

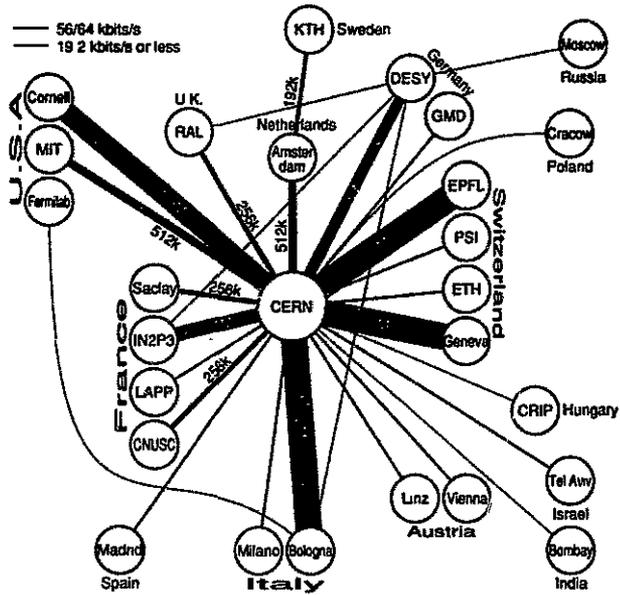


Figure 5. International network lines terminating at European institutes involved in experimental HEP. (CERN is treated as a separate country.)

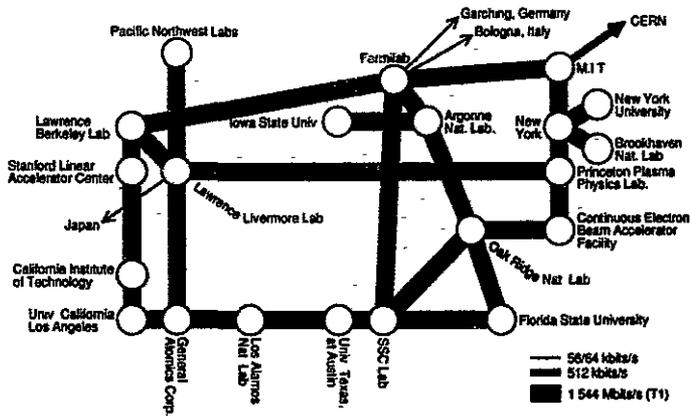


Figure 6. The Energy Sciences Network ESNET in the USA. Only the 'backbone' lines and international connections are shown. Approximately 40 lower speed lines to universities are not shown.

In most cases the 'policy' is to support any and all protocols for which there is any demand. For example, the recently installed  $768 \text{ kbit s}^{-1}$  line between CERN and

DESY provides DoD-IP†, DECNET-DDCMP‡, IBM-SNA§, X.25||, and ISO-IP¶ on separate channels with long-term bandwidth splitting determined according to usage. Along with the rising use of workstations and the increasing penetration of UNIX, HEP has seen a rapid rise in the demand for DoD-IP to support NFS††, FTP‡‡ and TELNET§§. The demand-led management of most of the network resources allows such changes to be accomplished without conflicts.

#### 5.4. The future of HEP networking

Most of the HEP network links that users consider adequate are funded directly by HEP agencies. Understandably, the management of a general-purpose network stressing connectivity, mail and remote logon services cannot always understand, and present strongly to funding agencies, the special needs of HEP. In addition, the management of a general-purpose network invariably tries to offer a minimum number of protocols, usually one, whereas HEP is forced by its international nature to support almost every protocol in existence.

Could HEP be served successfully by tomorrow's general-purpose networks? This seems unlikely during the next decade, but there are signs, particularly in commercial networking, that things are changing. The business world is poised to move to a largely electronic storage and transmission of images. This application, together perhaps with networked HDTV|||| on a slightly longer timescale, could make Gbit s<sup>-1</sup> networking routine (there are no technical problems apart from standardization issues). If there is a general demand for similar services and speed on academic networks, then the bandwidth, at least, of such networks might support HEP without difficulty.

We have left to last a development which may become very important. The prime purpose of HEP networking is to integrate university groups into a collaboration. Access to data at Mbyte s<sup>-1</sup> is necessary but is not the whole answer. Personal contact is vital, particularly for graduate students. As a pilot project, video teleconferencing has been installed in the USA between Lawrence Berkeley Laboratory on the West coast, Fermi National Accelerator Laboratory near Chicago, and the SSC Laboratory near Dallas. Several universities involved in the preparation of experiments for the SSC are now installing compatible equipment. The pilot project used 384 kbit s<sup>-1</sup> channels on the ESNET 1.544 Mbit s<sup>-1</sup> lines, but tests (Chartrand 1991) have demonstrated that current compression algorithms make 128 kbit s<sup>-1</sup> quite acceptable. The existing systems are really useful if several people in a few locations need to have a discussion. They are much less valuable for large meetings. Nevertheless, it is precisely the small, informal discussions that are lost to remote collaborators because the time and cost of travel is justified only for big meetings. Many funding agencies may still be pardoned

† The 'Internet Protocol' originally designed for use on the ARPANET set up by the US Department of Defense Advanced Research Projects Agency (DoD/ARPA).

‡ The 'Digital Data Communications Protocol' of Digital Equipment Corporation.

§ The 'Systems Network Architecture' protocols of IBM Corporation.

|| The protocols specified by recommendation X.25 of the Comité Consultatif International de Téléphonie et Télégraphie.

¶ The protocol, with functions very close to those of DoD-IP, specified by the International Standards Organisation.

†† The 'Network File System' of Sun Microsystems.

‡‡ The 'File Transfer Protocol' as specified by the DoD/ARPA.

§§ The virtual terminal protocol of DoD/ARPA.

|||| High-definition television.

for taking the view that anyone who wastes bandwidth on video teleconferencing must need a budget cut, but we predict that this will change completely over the next five years.

## 6. Computation and data handling

### 6.1. *The historical role of mainframes*

In the 1970s and early 1980s, large HEP laboratories invariably ran the fastest mainframes available. The use of these machines was justified by the undoubted need for their computing power, and by the fact that computing power was not available in any other form. With the appearance of vector 'supercomputers' came the realization that there might be different types of computer, but the vector hardware itself brought little benefit for HEP programs and the most appropriate mainframes for HEP appeared to be those designed principally for banks and insurance companies. Smaller 'supermini' computers were bought by university groups or individual experiments, but these machines cost almost exactly as much per MIPS as the mainframes.

About 1976 a few small groups realized that, although the mainframes were very well suited to the large fraction of I/O intensive computing, there were some tasks which could be done faster with a supplement of cheap CPU power. These pioneers built the 'emulators' (Kunz 1976, Halatsis 1980, Kunz 1984, Brafman and Notz 1984, Ferran *et al* 1985) where each emulator was a crate containing a few cards and could execute code compiled on a mainframe at a good fraction of the mainframe's speed. Emulators were rather tedious to use, but a number of experiments made heavy use of them to perform otherwise unthinkable long tasks.

### 6.2. *The RISC revolution*

By 1985 it was clear to the majority of physicists that, unless they were prepared to suffer considerable inconvenience, there was no way to get really cheap computing. Few expected reduced instruction set computers (RISC) to improve cost effectiveness significantly. By 1988, when L3 ordered its first Apollo DN10000, the majority (including the author) had been proved wrong. Whether the root cause was RISC, or just the economic benefits of squeezing a complete CPU onto one or two chips, is the subject of continuing debate but the question is unimportant for HEP. Figure 7 shows the fall with time of the relative cost (\$/HEP-MIPS) of IBM mainframes, emulators and RISC workstations.

Could a mainframe be replaced by RISC workstations connected by a local area network (LAN) thus saving a lot of money? One problem with this approach is that HEP needs devices such as tape robots and 'computer-centre-quality' peripherals which are still mainly available on mainframes. A second problem is the network bandwidth limitations discussed below. A third problem is the need for sophisticated system-wide resource management which is routine on mainframes but absent or primitive in workstation clusters. Despite these problems, it is now common for over 90% of the MIPS required by an experiment to be provided by RISC systems, even if a substantial amount of money is still spent on more traditional equipment.

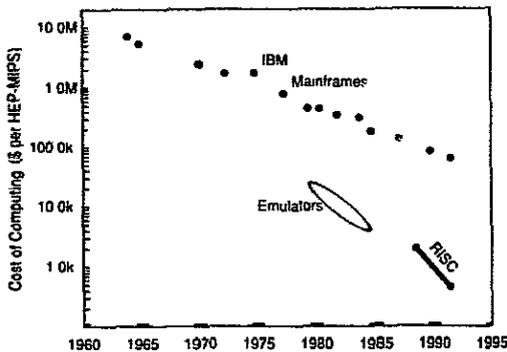


Figure 7. The variation with time of the relative cost of IBM mainframes, emulators and RISC workstations. The IBM mainframe data for 1964 to 1985 were taken from *Datamation*, 15 May 1985.

### 6.3. Local area networks and links

ETHERNET was first described in 1976 (Metcalfe and Boggs) and became a usable standard in the early 1980s. The first ETHERNET interfaces and software supported communication at a few percent of the 10 Mbit  $s^{-1}$  nominal bandwidth; only within the last few years has it become possible to exchange data at a substantial fraction of 10 Mbit  $s^{-1}$ . The equivalent of ETHERNET for the 1990s is almost certainly FDDI† which is already in limited use in HEP providing, typically, 10% of its nominal 100 Mbit  $s^{-1}$  bandwidth. LAN speeds have been rising much more slowly than RISC MIPS/\$. ETHERNET is quite inadequate to support data-intensive physics analysis on a cluster of 10–50 MIPS workstations, as is daily demonstrated by several existing HEP clusters. FDDI will help a little but will still be too slow.

The advantage of ETHERNET (and later of FDDI) is that this interface is supplied with almost every workstation. In other words, ETHERNET is a widely accepted standard. The disadvantage with widely accepted standards is that they lag far behind technology. To take advantage of technological possibilities to meet the real communications needs, HEP has had to leave the world of widely accepted standards. Until recently this meant either buying vendor-specific equipment, or using home-made or barely commercial interfaces to interconnect heterogeneous equipment. In the last two years, standards for high-performance links have begun to emerge. HIPPI‡, a parallel copper link with 100 Mbyte  $s^{-1}$  throughput, and the less mature FCS§, offering similar performance over optical fibres, are the subject of intense work by ANSI. HIPPI interfaces will never be standard on every workstation, but they (and/or FCS) may be optional extras which HEP could exploit for its unusually high communications needs.

### 6.4. Data-handling technology

While the RISC revolution was a surprise, the slower evolution of mass storage technology summarized in table 3 has been a disappointment. 'Computer-centre-quality' magnetic tapes holding 160 Mbyte became widely used in HEP from about 1976.

† Fiber Distributed Data Interface, ANSI standard.

‡ High-Performance Parallel Interface, ANSI Standard (Standards Committee X3T9.3).

§ FibreChannel Standard, ANSI Draft Standard (Standards Committee X3T9.3).

Table 3. Typical mass storage costs for HEP in \$/Gbyte. The costs for tapes include media and drives in an appropriate proportion for HEP. The costs for disks include the storage controller (for mainframes) or the host workstation (for workstation file servers). Where estimates are given for 1992-3 they are based on recently announced products not yet widely used in HEP.

	1982	1992 (now)	1992-3
Computer centre manually mounted tapes	140	50	13
Computer centre robot-mounted tapes	—	240	60
Helical scan tapes (8 mm)	—	15	6
Mainframe disks	35 000	5000	—
Workstation disks	70 000	2700	2200
Optical disk 'juke box'	—	1200	—

In 1990, the equivalent magnetic tape cartridge had a capacity of only 200 Mbyte, although it was less bulky.

Computer centre magnetic tapes are limited in capacity less by technology than by the small perceived market for mass storage. Helical scan devices, mainly using hardware developed for video and audio entertainment products show that 2-5 Gbyte can be stored in a small cartridge (8 mm video or DAT†). Larger helical data recorders, for example from Sony and Ampex, can record 200 Gbyte on a cartridge weighing several kg. Particularly in the USA, 8 mm drives sold by Exabyte are used extensively for recording and distribution of data which will probably be read only once; when such tapes are read or written repeatedly, reliability decreases unacceptably.

The current generation of computer centre tapes and drives is IBM 3480 compatible. Physically, a 3480 cartridge is about 12 cm square, weighs 230 g, and has a capacity of 200 Mbyte which by the end of 1992 will have risen to 800 Mbyte. The design of 3480 cartridges and drives produces very high data integrity—the only major loss of 3480 data belonging to the L3 experiment involved a cartridge which was dropped and run over by a tape-mounting robot!

There is no clear technical reason why workstation magnetic disk storage should have improved in capacity/cost faster than magnetic tape or mainframe disk; this anomaly is probably due to the fiercely competitive market for small (~ 1 Gbyte) disks. Using these components many vendors are now producing 'redundant arrays of inexpensive disks', which appear as high-capacity, high-speed, high-reliability (but rarely inexpensive) storage devices.

Optical storage has been 'just round the corner' for a decade. Although the latest devices may have some commercial success, they do not meet the needs of HEP. Viewed as bulk storage, the media are too expensive and the drives are too slow in comparison with magnetic tape. Viewed as random access storage, an optical 'juke box' with, for example, 4 drives and 100 disks, is unsatisfactory for more than four simultaneous users and is not vastly cheaper than magnetic disks.

### 6.5. A model for HEP computing

Workstation CPUs will continue to increase in speed by a factor of two every 18-24 months for most of the 1990s (at least). General-purpose networks will increase in speed much more slowly. We must accept that workstations linked to each other

† Digital Audio Tape.

and to data server(s) by general-purpose networks alone will become increasingly isolated—we will be unable to gain full benefit from the increasing number of MIPS in each box.

Figure 8 shows conceptually how high-speed links can be added to general-purpose networking to build a workstation-dominated computing system in which the bandwidth needs are met (Mount 1991). For simplicity, figure 8 shows only the limited number of workstations performing data-intensive computing and omits the many desktop machines and the CPU-intensive simulation facility that are also required. This model is increasingly accepted as an appropriate starting point for designing the computing systems for experiments at the SSC and LHC. The big question unanswered by figure 8 is 'what is the data server?'. Today, a full-function data server has to include a mainframe because these are the machines with access to the bulk of the data. In the future there is certainly no reason for the data server to be called a mainframe, but the tightly interconnected high-quality peripherals and data-management system needed to provide the data-server function will probably retain a remarkable similarity to whatever the banks and insurance companies are buying to meet their needs.

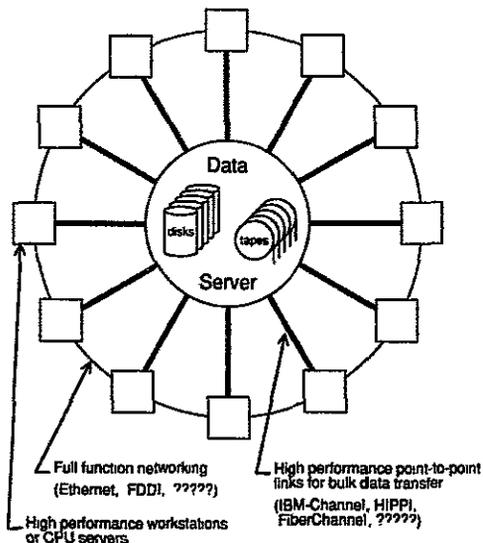


Figure 8. A model for HEP computing. The data server and the most powerful workstations are shown.

The model is not just a logical deduction from the evolution of CPU and network performance; it is also an idealized and simplified summary of several existing HEP computing systems. Figure 9 shows the distributed computing system which has been built by and for the L3 experiment. Most of L3's 32 DN10000 and 90 other workstations are omitted from the figure. General-purpose networking is provided by FDDI, Ethernet and Apollo Token Ring ( $12 \text{ Mbit s}^{-1}$ ) and high-speed ( $1\text{--}2 \text{ Mbyte s}^{-1}$ ) links are provided by 'Stollmann' interfaces linking DN10000s to the IBM 3090. The IBM has access to thousands of robot-mounted tapes and tens of thousands of manually mounted tapes. More details of 'EPICS, the hub of the L3 system, are given in

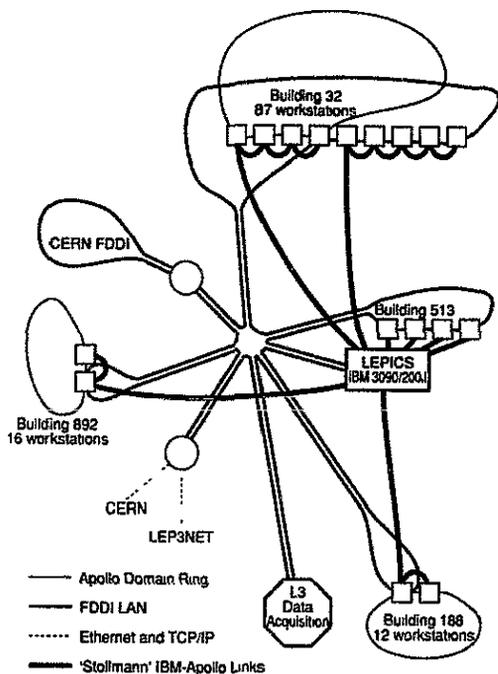


Figure 9. The computing systems, networks and links used by the L3 experiment at CERN. Distances between buildings range from 1.5 to 10 km.

section 6.8.1.

Figure 10 shows a second example of a real system, of which figure 8 is a simplified model. SHIFT, the Scalable Heterogeneous Integrated Facility for HEP Computing, (Baud *et al* 1991) uses the ULTRANET† proprietary system to interconnect workstations and data-server facilities at aggregate speeds of up to  $1 \text{ Gbit s}^{-1}$ . The Cray shown in figure 10 is not a dedicated component of SHIFT, but as one of its tasks, it provides the high-speed access to tape-resident data which is vital to SHIFT's role as a system supporting data analysis.

The model also has some similarity to the view of the whole CERN Computer Centre (of which SHIFT is a part) shown in figure 11. It should be noted that the workstation-based simulation facility only requires limited access to data. In contrast, the workstation-based analysis facility, SHIFT, needs good access to data, provided partly by direct access to the tape vault and partly by ULTRANET links to the Cray and IBM systems.

### 6.6. Vector and parallel computing

That most HEP programs are not naturally vectorizable has already been mentioned in section 2.2. Vector hardware can be very useful to accelerator physicists and for some specialized HEP uses such as the calculation of a 3D magnetic field map, but has little impact on the bulk of the computing load.

† Ultra Network Technologies Inc.

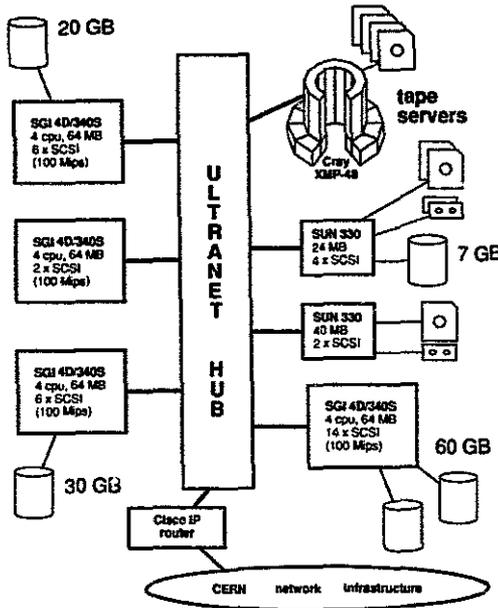


Figure 10. The SHIFT workstation-based data-analysis facility at CERN.

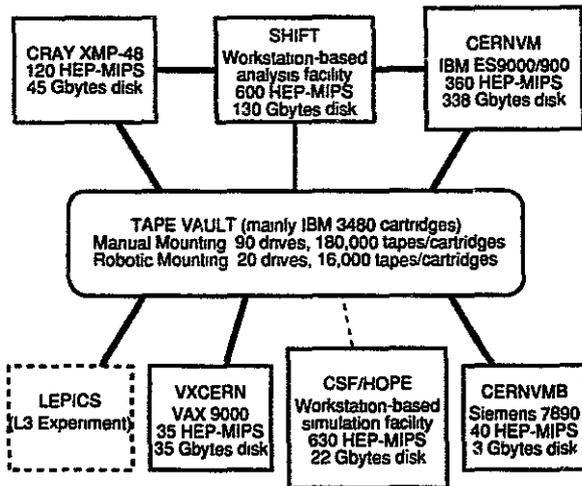


Figure 11. A simplified view of the CERN Computer Centre showing the principal facilities. (Note that CERNVMB has access to CERNVM disks.)

In contrast, most HEP computing exhibits an obvious parallelism. Individual events, which we firmly believe to be independent of each other, can be treated in any order or all at once. At the most trivial level, a large HEP task can always be subdivided into  $N$  batch jobs which are run on  $N$  computers to achieve a reduction in processing time very close to  $N$ . A more elegant approach is to run a single parent job which feeds events to child processes each running on a separate processor.

Such systems, often known as farms, are used by most large experiments for their more routine processing. Although in many ways trivial, this parallel computing still requires considerable software effort to give child tasks access to the experiment's database and to bring together the diagnostic and statistical information provided by each process. Thus the less routine tasks in physics analysis are rarely, if ever, parallelized.

Several compilers for multiprocessor computers will now attempt to find, automatically, some of the parallelism in the code. Typically this is successful for simple loop structures which are entirely within one subroutine. In contrast, HEP's natural parallelism could be discovered only by considering hundreds of thousands of lines of code spread over hundreds of subroutines. It appears that HEP will have to edit parallelism by hand for several decades at least.

### 6.7. *Software support for parallel computing*

The manual approach to parallelism still needs appropriate HEP and system utilities. Parallelism in HEP is worthwhile only if it reduces, or keeps low, the cost of computing. Cheap parallel systems are assembled from cheap components and inevitably lack the high-bandwidth access to shared memory assumed in many models of parallel computing. Thus an appropriate model for HEP parallel computing assumes that processors communicate by send/receive at any time during process execution, and that the amount of data sent is under application control. Appropriate sets of subroutine calls to support this type of parallelism have been described and implemented by several HEP groups (e.g. Biel 1987, May 1989, Dittus 1989, Kaliher 1990). Stimulated by joint projects with HEP, at least one major manufacturer has provided or improved parallel processing support under FORTRAN to meet these needs. However, the solutions, like those created up to now within HEP, have always been specific to a very limited range of hardware platforms. An industry-based initiative, which began as the 'Parallel Computing Forum' has now become an subcommittee of the ANSI committee X3.H5; their draft standard defining parallel processing support within high-level languages will soon be released for public review.

Within the academic world there appears to be a wide consensus that, in the longer term, loosely coupled parallel processing should be supported via the interprocess communications facilities provided by BSD† 4.3 sockets. These sockets are commonly implemented over TCP/IP, which being designed for wide-area communications, can limit performance in some environments. Efficient BSD socket implementations over HIPPI, FCS and future standardized links will be vital to the success of this approach. Even assuming standardization on BSD sockets, much additional software is required and eventual convergence with the ANSI X3.H5 standard is not ruled out.

### 6.8. *Examples of HEP computing systems*

The CERN Computer Centre already shown in figure 11 is typical of the computing centres to be found at HEP laboratories. The key features are a central tape/cartridge library including a robotic component, the continued importance of general-purpose mainframe processors, and the rising importance of workstation-based systems. Details of the central installations at each laboratory are available in several reviews (e.g. Cooper 1990, Mount 1990, Amako 1991) and will not be repeated here. However, one central service deserves special comment. The SSC Laboratory runs a central

† The version of UNIX distributed by the University of California, Berkeley.

computing service for physicists which is based entirely on RISC workstations (Cornell 1991). Since the SSC will not generate real collisions until the end of the century, the principal function of its central service is to run the detailed event and detector simulations needed in the design of the SSC experiments. These functions can be effectively provided by a farm (or in Texas a ranch) of (mainly screen-less and keyboard-less) RISC workstations. The absence of even a scaled-down equivalent of the CERN tape vault is, at this stage in the life of the SSC Laboratory, only a mild inconvenience.

In the next part of this section we will examine three of the parallel computing systems in current use, paying particular attention to the extent to which their construction has been justified.

**6.8.1. Examples of parallel computing systems in HEP.** The three examples of parallel computing systems in current HEP use range from an assembly of products from a single manufacturer, to a more ambitious system which has been specially designed from the board layout upwards. That the parallelism in the first example (at least) is far too trivial to be of interest to computer scientists is of no concern at all to its HEP users.

(i) *The FALCON system for the ALEPH experiment.* Figure 12 shows the system used by the ALEPH experiment at the LEP accelerator to reconstruct events almost immediately after data acquisition. The figure shows the first version of this system which was assembled before LEP data taking started in 1989. Data are acquired from the detector by a VAX (not shown in the figure) and buffered on dual ported disks. Event reconstruction is performed on the 12 diskless, screenless VAXstation 3100/30 processors producing compact output with a size about one third that of the raw events. The output is transmitted over ETHERNET to CERN's IBM 3090 system for storage and further analysis.

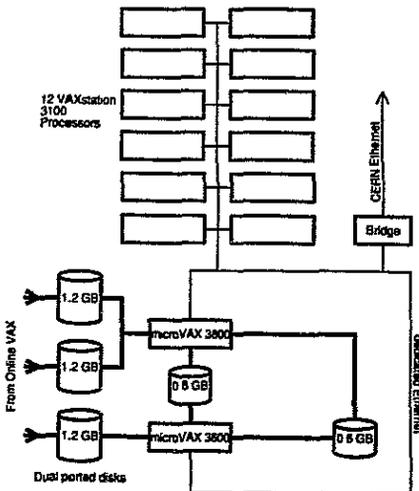


Figure 12. The FALCON parallel event reconstruction system used by the ALEPH experiment (1989–1990 configuration).

FALCON is composed entirely of 'off-the-shelf' hardware from Digital Equipment Corporation. Its CPU power is modest (rising from 24 HEP-MIPS in 1989 to 90 HEP-MIPS now through the use of more recent VAXstations), it contains no RISC processors and its total bandwidth for interprocessor communications is limited by its dedicated ETHERNET. However it has worked without problems from the beginning of LEP data taking and has been able to keep up with ALEPH data in real time (Delfino *et al* 1991).

(ii) *The LEPICS system for the L3 experiment.* Figure 13 shows the 'L3 Parallel Integrated Computer System' (LEPICS) which comprises an IBM 3090 mainframe coupled to a number of HP/Apollo DN10000 RISC workstations using interfaces made by Stollmann GmbH. The attached processors can either run parallel reconstruction processing or other simulation or analysis tasks. Communication between the 3090 and the DN10000s is used both for parallel processing and to support access from the workstations to mainframe-resident data. The CPU power of LEPICS totals 240 HEP-MIPS and will rise to about 500 HEP-MIPS as additional interfaces are used to connect existing DN10000s.

Like FALCON, LEPICS was physically assembled before data taking started in 1989. However, the parallel-processing software was not fully operational until the end of 1990. Although there is a detailed technical explanation for this delay, perhaps the simplest explanation is that assembling a system composed of two major vendors' hardware, which communicate using a third-party interface for which drivers had to be written, was a bigger task than anticipated. The task would have taken even longer were it not for joint projects with IBM, on parallel processing, and with Apollo, on the driver for the Stollmann interface.

(iii) *The ACP/R3000 system at Fermilab.* Figure 14 shows an example of an ACP/R3000 system. Fermilab's 'Advanced Computing Project' (Nash *et al* 1983, Gaines *et al* 1987, Nash 1989) began at a time when mainframes still appeared to dominate. The first ACP systems were composed of single board, 0.7 HEP-MIPS computers in VME crates. The ACP boards acted as attached processors to a host computer as in most simple HEP 'farms'. Over 500 of the original ACP boards were built and they performed a vast amount of processing. However, they were only considered suitable for very stable software, since the FORTRAN cross-compiler for the ACP processors had many difficulties with large programs making software installation very time consuming.

The second-generation ACP system uses the R3000 family of processor chips from MIPS Corporation for which a robust FORTRAN compiler has long been available. Each processor can have from 8 to 32 Mbyte of memory and has a CPU power of 10–12 HEP-MIPS. The ACP/R3000 departs from the familiar host/attached processor concept: the system can be logically partitioned so that an appropriate number of processors perform each step in even processing, from the initial reading of an input tape through to the writing of the output tape. When a processor has completed its work on an event, the event is passed to one of the processors in the pool responsible for the next stage of processing. Over 200 ACP/R3000 processors have been built and systems have been running at Fermilab and some universities since 1991.

**6.8.2. Verdicts on HEP parallel systems.** It must be clearly understood that verdicts on FALCON, LEPICS and ACP-II can only reflect the subjective judgment of the author.

FALCON is an undoubted success. It performs a very specialized computing task at a small fraction of the cost of employing a mainframe. The manpower needed to get

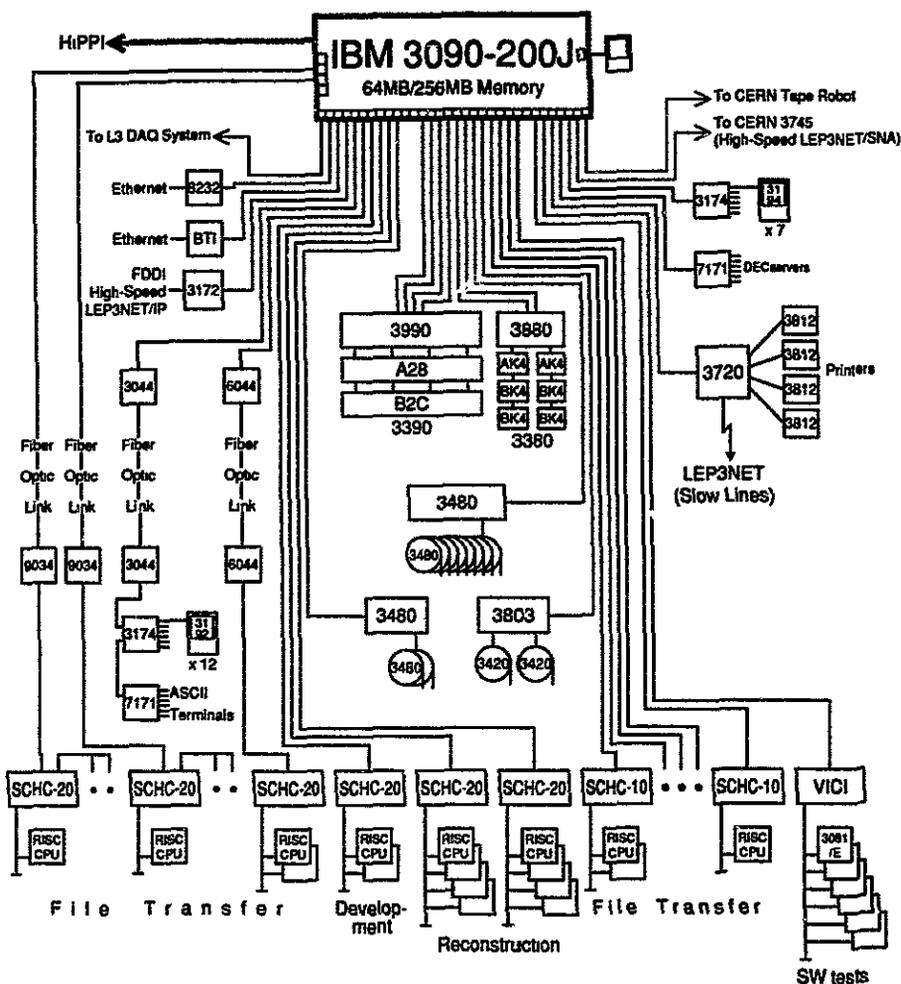


Figure 13. The LEPICS system for the L3 experiment at LEP as it will be after the completion of upgrades now in progress. The attached RISC processors are HP/Apollo DN1000s.

the FALCON system running was modest and keeping it running appears to be simple.

The complete LEPICS system, mainframe plus workstations, works well and is a very flexible computing resource. It is, in many ways, a success, but L3 was inconvenienced by the delays in making parallel processing work, and finding manpower to maintain and develop the specialized software is not easy. The message to a future large experiment is 'don't take on anything more demanding than LEPICS'.

The ACP/R3000, while not as specialized as FALCON, is not a general-purpose HEP computing system. It has, however, required a substantial investment of Fermilab manpower which would have been quite unthinkable for a single experiment. ACP/R3000 systems are heavily used, but even at Fermilab, the acquisition of a collection of the fastest RISC workstations at a good discount is often preferred to buying

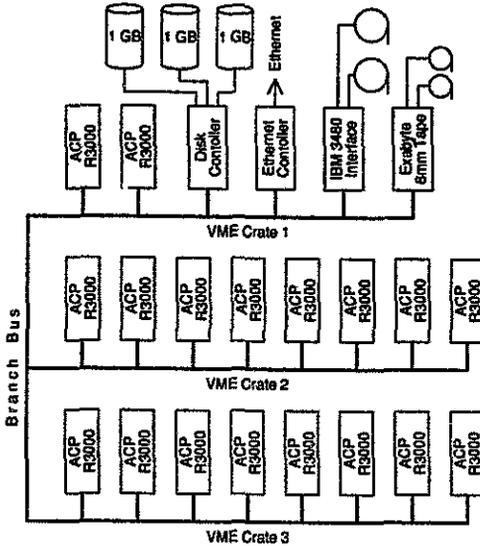


Figure 14. The ACP/R3000 system at the University of Michigan. Each ACP/R3000 processor has 16 Mbyte of memory.

ACP/R3000 equipment at the cost price of replication. Perhaps the verdict on the ACP/R3000 should be that we can normally expect to be able to buy our CPU power from industry at the last moment, but to put ourselves in a position where we have no alternative would be irresponsible.

*Commercial parallel processing systems for HEP* Within the last few years, two of the world's largest computer manufacturers built prototypes of parallel computing systems reasonably well matched to HEP needs. One complete prototype from one vendor was installed at CERN for tests, and components of the other were used for software development in preparation for the arrival of a complete machine. As it happened, neither system was considered by its makers to be sufficiently widely marketable and both disappeared. There seems to be no reason for the logic of the marketplace to change rapidly. Thus we expect that HEP systems will continue to be built by HEP, even if we use readily available workstations and peripherals as components.

## 7. Conclusions

High-energy physics experiments are among the world's most demanding users of computing systems, especially in their simultaneous need for large-scale data handling and high computational power. The high statistical precision needed to make precise measurements in a quantum world means that high-energy physicists must always strive to build experiments which will have large computing needs. A desire, perhaps a need, to keep HEP a science dominated by universities brings a requirement for affordable, high-bandwidth wide-area networks.

The analysis of data from a large HEP experiment requires that physicists write hundreds of thousands of lines of computer programs. Formal or less formal software engineering techniques, including self-discipline, have made major contributions

towards allowing these programs to be used in precise physics studies. The core of well-tested software packages developed within HEP provides a further aid to reliable analysis. Finally, the advent of the workstation environment for program development and interactive graphics has allowed physicists to understand, much better than before, what their software is really doing.

High-energy physics has been a leader in moving appropriate applications to more cost effective systems than mainframes and supercomputers. Where necessary, specialized computing systems and interfaces have been designed and implemented. It is already common for about 90% of computation to be performed on workstations or specially built processors rather than on mainframes. However mainframes and their peripherals are still used extensively for large-scale data handling.

Towards the end of the century, proton-proton colliding-beam experiments at 16-40 TeV will need systems with computing power in the region of 1 000 000 MIPS and the ability to handle thousands of terabytes of data. It is unlikely that adequately cost-effective systems will be available off-the-shelf from the computing industry, but it is quite possible that appropriate systems could be assembled using readily available components and interconnections. The availability of standard high-speed interconnections and the software to support their exploitation will be a key to meeting the special needs of HEP. In all areas of computing, but especially in connection with high-speed links, it will be necessary to avoid the evils, either of being unprepared, or of investing a vast effort in building hardware or software for which perfectly adequate substitutes become commercially available. The evils are best avoided by close contacts with the computing industry and by using joint industry-HEP projects to develop needed hardware or software which might not otherwise become available in time.

### Acknowledgments

I would like to thank my many colleagues who have discussed their work with me and have thus knowingly or unknowingly contributed to the material presented in this review. I am particularly indebted to R Ball, F Fluckiger, E Jagel, J Knobloch, H Kowalski, K Mcfarlane, O Martin, U Martyn, H Renshall, D Underhill, C Youngman for supplying me with information and/or figures and to my wife Jane for help in preparing and checking the manuscript.

My treatment of individual projects and achievements has been necessarily very limited. I recommend that any reader seeking a better understanding of HEP computing turn first to the proceedings of the series of international conferences on computing in high-energy physics (*CHEP* 1984 to *CHEP* 1991).

### References

- Abbiendi G and Stanco L 1991 *Comput. Phys. Commun.* 66 16-24
- Adeva B *et al* 1991a *Nucl. Instrum. Methods A* 309 318-30
- 1991b *CHEP (Computing in High Energy Physics Conf. Series)* pp 323-8
- Anako K 1991 *CHEP (Computing in High Energy Physics Conf. Series)* pp 11-21
- Baud J-P *et al* 1991 *CHEP (Computing in High Energy Physics Conf. Series)* pp 571-82
- Bengtsson H-U 1984 *Comput. Phys. Commun.* 31 323-55
- Bengtsson H-U and Sjöstrand T 1987 *Comput. Phys. Commun.* 46 43-82

- Biel J *et al* 1987 *CHEP (Computing in High Energy Physics Conf. Series)* pp 331-7
- Biobel V 1988 *DESY Internal Report* R1-88-01
- Brafman H and Notz D 1984 *CHEP (Computing in High Energy Physics Conf. Series)* p 211
- Brun R and Carminati F 1991 *CHEP (Computing in High Energy Physics Conf. Series)* pp 451-8
- Brun R and Zoll J 1987 *ZEBRA user guide* CERN Program Library Q100
- Brun M, Brun R and Rademakers A 1989a *CHEP (Computing in High Energy Physics Conf. Series)* pp 235-8
- Brun R, Bruyant F, Maire M and McPherson A 1989b *GEANT Long Writeup* CERN Program Library W5103
- Brun R, Couet O, Vandoni C and Zanarini P 1989c *CHEP (Computing in High Energy Physics Conf. Series)* pp 432-7
- P 1989d *PAW-Physics Analysis Workstation, The Complete Reference, Version 1.07* CERN Program Library Q121
- Brun R, Hansroul M and Lassale J C 1984 *ZBOOK Long Writeup* CERN Program Library Q210
- Catani S, Marchesini G and Webber B 1991 *Nucl. Phys. B* 349 635-54
- Cavallini J *et al* 1991 *ESNET Program Plan* (Washington: US Department of Energy)
- Chartrand G 1991 *CHEP (Computing in High Energy Physics Conf. Series)* pp 703-8
- CHEP 1984 *Proc. Symp. Recent Developments in Computing, Processor and Software Research for High Energy Physics (Guanajuato, Mexico, 8-11 May 1984)* ed R Donaldson and M Kreister (Mexico City: Univ. Nac. Auto. Mexico)
- CHEP 1985 *Proc. Conf. on Computing in High Energy Physics (Amsterdam, The Netherlands, 25-28 June, 1985)* ed L O Hertzberger and W Hoogland (Amsterdam: Elsevier)
- CHEP 1987 *Proc. Int. Conf. Computing in High Energy Physics (Asilomar, USA, 2-6 February, 1987)* ed W Ash (*Comput. Phys. Commun.* 45)
- CHEP 1988 *Proc. Int. Conf. on the Impact of Digital Microelectronics and Microprocessors on Particle Physics (Trieste, Italy, 28-30 March 1988)* ed M Budinich, E Calstelli and A Colavita (Singapore: World Scientific)
- CHEP 1989 *Proc. Int. Conf. on Computing in High Energy Physics (Oxford, UK, 10-14 April, 1989)* ed R C E Devenish and T Daniels (*Comput. Phys. Commun.* 57)
- CHEP 1990 *Computing for High Luminosity and High Intensity Facilities (Santa Fe, USA, 9-13 April, 1990)* (*AIP Conf. Proc.* 209) ed J Lillberg and M Oothoudt (New York: AIP)
- CHEP 1991 *Computing in High Energy Physics '91 (Tsukuba, Japan, 11-15 March, 1991)* ed Y Watase and F Abe (Tokyo: Universal Academy Press)
- Cooper P 1990 *CHEP (Computing in High Energy Physics Conf. Series)* pp 3-15
- Cornell L 1991 *CHEP (Computing in High Energy Physics Conf. Series)* pp 77-83
- Cranfield R, Holl B and Jones R W L 1991 *OPAL User Guide* OPAL Collaboration, CERN, OC504//OPAL/OFFL/36/0003
- Delfino M, Pacheco A and Knobloch J 1991 *CHEP (Computing in High Energy Physics Conf. Series)* pp 177-81
- Dittus F 1989 *CHEP (Computing in High Energy Physics Conf. Series)* pp 395-400
- Erice-1990 1991 *Proc. 14th INFN Eloisatron Project Workshop for Particle Physics Experiments: Data Structures for Particle Physics Experiments (Erice, Italy, November 1990)* (Singapore: World Scientific)
- Ferran P M *et al* 1985 *CHEP (Computing in High Energy Physics Conf. Series)* pp 322-5
- Fesefeldt H 1985 *The Simulation of Hadronic Showers, Physics and Applications III* Physalisches Institute, RWTH Aachen PITHA 85/02
- Fluckiger F 1991 *CHEP (Computing in High Energy Physics Conf. Series)* pp 709-17
- Gaines I *et al* 1987 *CHEP (Computing in High Energy Physics Conf. Series)* pp 323-9
- Halatsis C *et al* 1980 *Comput. Architecture News* 8 278-91
- Jadach S and Was Z 1985 *Comput. Phys. Commun.* 36 191-211
- Jadach S, Wård B F and Was Z 1989 *Z Physics at LEP1 (CERN Yellow Report 89-03)* ed G Altarelli, R Kleiss and C Verzegnassi
- Kalisher C 1990 *CHEP (Computing in High Energy Physics Conf. Series)* pp 364-71
- Klein H and Zoll J 1980 *BATCHY users guide* CERN Program Library LA00
- Knowles I G 1988 *Nucl. Phys. B* 310 571-88
- Kunz P F 1976 *Nucl. Instrum. Methods* 135 435-40
- 1984 *CHEP (Computing in High Energy Physics Conf. Series)* pp 197-209
- Lidinsky W 1991 *CHEP (Computing in High Energy Physics Conf. Series)* pp 719-28
- Lyon-1990 1991 *Proc. First Int. Workshop on Software Engineering, Artificial Intelligence and Expert Systems*

- in *High Energy and Nuclear Physics (Lyon, France, 19-24 March, 1990)* ed D Perret-Gallix and W Wojcick (Paris: Centre National de la Recherche Scientifique).
- Marchesini G and Webber B 1988 *Nucl. Phys. B* **310** 461-526
- Marchesini G, Webber B, Abbiendi G, Knowles I G, Seymour M H and Stanco L 1992 *Comput. Phys. Commun.* **67** 465-508
- May E 1989 *CHEP (Computing in High Energy Physics Conf. Series)* pp 278-84
- Metcalfe R and Boggs D 1976 *Commun. ACM* **19** 395
- Mount R 1990 *CHEP (Computing in High Energy Physics Conf. Series)* pp 44-59
- 1991 *CHEP (Computing in High Energy Physics Conf. Series)* pp 691-701
- Nash T 1989 *CHEP (Computing in High Energy Physics Conf. Series)* pp 47-53
- Nash T *et al* 1983 Fermilab's Advanced Computer Research and Development Program *Fermilab Report FN 83*
- Nelson W W, Hirayama H and Rogers D W O 1985 *The EGS4 Code System* Stanford Linear Accelerator Center SLAC-265
- Paige F and Protopopescu S 1986 ISAJET 5.30: a Monte Carlo event generator for pp and  $p\bar{p}$  interactions *Brookhaven National Laboratory Report BNL 38774*
- Shiers J 1991 *CHEP (Computing in High Energy Physics Conf. Series)* pp 329-33
- Shiers J and Goossens M 1991 *FATMEN Distributed File and Tape Management System* CERN Program Library Q123
- Sjöstrand T 1982 *Comput. Phys. Commun.* **27** 243-84
- Sjöstrand T and Bengtsson M 1987 *Comput. Phys. Commun.* **43** 367-79
- Sjöstrand T and van Zijl M 1987 *Phys. Rev. D* **36** 2019-41
- White B 1989 *CHEP (Computing in High Energy Physics Conf. Series)* pp 538-42
- Zoll J 1989 *HRDR Long Writeup* CERN Program Library Q101