

## Chapter 9

### Data Handling and Communication

Frédéric Hemmer and Pier Giorgio Innocenti

#### 9.1 Introduction

Accelerators and experiments in high energy physics cannot be conceived today without the extensive use of computers. For accelerators, computers are used for the simulation of the performance, and together with data communication for the operation, control and performance monitoring. For the experiments they are essential for simulating the expected performance, selecting the events of interest and analysing the data. Data volumes are generated in all these steps and are very large, requiring careful attention to data storage and data retrieval, as well as to efficient programming. Digital communication is needed for cooperative working between processors and between the physicists.

This chapter traces the evolution of computer usage at CERN, with an emphasis on the impact on experimentation, from SC to LHC [1].

In 1958 the computer age started (Fig. 9.1) at CERN with the operation of the Ferranti Mercury, a machine based on vacuum tubes [3]. It was used for calculations by engineers and physicists and for processing the data produced by the scanning and measurement of pictures (recorded on photographic film) obtained by bubble chambers and optical spark chambers.

#### *Processing of bubble chamber pictures*

The arrival of the IBM 709 in 1961 marked a progress in the processing of bubble chamber and spark chamber pictures. While still using vacuum tubes, it had for the time modern peripherals (card punches, tabulators and magnetic tapes). Importantly, it used FORTRAN as a programming language, in which the first general purpose programs for track reconstruction and kinematics were written. Two years later, the IBM 7090 arrived, based on transistor electronics and with good input/output facilities.

Digitizers for bubble chamber and spark chamber pictures were linked on-line to the computer to speed up the input process, a substantial improvement over transfer by paper tape; processing of data from film represented the main usage at that time.

Remote access to the central computers became available in the late 1960s [Highlight 9.3].

The era of central mainframe computers lasted for more than thirty years. The CERN Computer Centre operated a number of IBM machines, the CDC 6600 and 7600 and a Cray X-MP supercomputer. Given the demanding applications and the knowledge of its staff, CERN was a highly appreciated partner for the computer manufactures, who were relying on CERN experience to improve their products.

The FORTRAN “culture” linked to bubble chamber data processing produced a library of programs used worldwide by the high energy physics community; this approach was extended to processing of data produced by electronic experiments. Advanced code management methods were developed [Highlight 9.6].

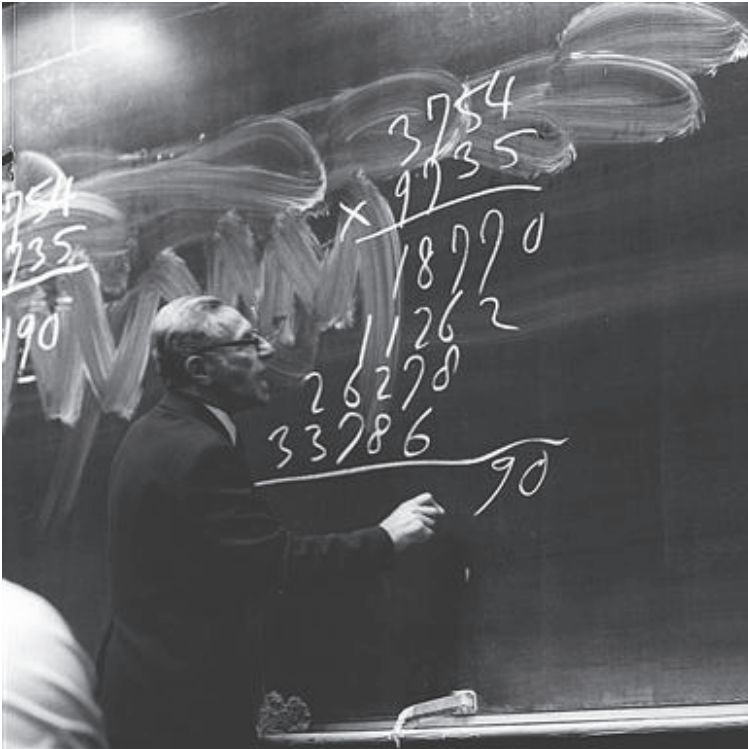


Fig. 9.1. Wim Klein, the first CERN computer, was operational in the Theory Division before the Ferranti Mercury [2].

## ***Electronic experiments***

In 1963 the appearance of the spark chambers with electronic readout prompted radical change in the use of computers. Before that time, electronics experiments, as opposed to experiments using photographic film, manually recorded event rates and logical combinations of signals, such as time coincidences, produced by detectors with coarse space resolution. The novelty of the digitized spark chamber was that, after a trigger, it produced prompt hits (electronic pulses) of particles in the detectors, with good space resolution; the hits were suitable for immediate digitization and recording by a computer.

The data acquisition for the Missing Mass Spectrometer [4], tested at the SC in 1963 with acoustic spark chambers, displayed all the features of a modern particle physics experiment: a selective fast trigger, fine space resolution, a digitizer for each data channel, a link [5] to a computer which recorded the event on tape, performed full reconstruction and displayed the results. The computer was the Ferranti Mercury: It had a relatively slow input capability, but its high processing power permitted the complete reconstruction and analysis of one out of ten events recorded on tape. Immediate display of the results from the sample offered a novel valuable monitoring of experiment performance.

The Ferranti Mercury ended its useful life with that experiment. Minicomputers using solid state electronics became popular in data acquisition in the sixties, but could not perform a complete analysis of a data sample, due to their limited processing power and to the lack of a FORTRAN compiler: FORTRAN had become the *de facto* standard language for data analysis.

In the early 1970s, the deployment of multiwire proportional chambers [Highlight 4.8] in many experiments brought a large increase in data volume: These detectors have typically many readout channels to achieve good space resolution and have high rate capability. Processing a large number of complex events, such as those produced on fixed target at the SPS or by colliding beams at the ISR, required innovative software methods. This trend towards more complex events, each consisting of a large data set, has continued, with a notable increase due to the very fine grain fast silicon detectors, culminating at the LHC.

With the increasing complexity of the detectors and data selection criteria, computers became part of the trigger strategy. Triggers, earlier obtained from timing coincidences and Boolean logic, now used simple hardware processors and computers performing calculations as required for instance by pattern recognition. The minicomputers available at the time had marginal computing power for this application, and required lengthy re-programming and adaption of the algorithms used in the full off-line reconstruction.

A solution came from the use of special processors (370/168E) developed at the Stanford Linear Accelerator Centre (SLAC) [6]. Running on custom designed hardware, they emulated a reduced set of the instructions of the IBM 370 mainframe, which were needed to execute the reconstruction programs used in the program libraries: the available reconstruction programs could then be used with only minor modification. One such processor was used as third level trigger to select a high transverse momentum particle in an experiment with the Split Field Magnet detector at the ISR [7] in the late 1970s. Clusters of 168E and of 3081E (emulating the IBM 370/3081) were extensively used by the UA1 trigger [8] and off-line reconstruction, and by other experiments.

The introduction of emulators was the first successful example of a full precision arithmetic algorithm applied to real time selection of events. These developments, favoured by the availability of affordable large computing power, started a slow evolution of data taking, from recording the raw hits in the detectors after a coarse selection, to putting to permanent storage only the events selected by a complete pattern recognition carried out in real time. In this mode of operation, the reliability of trigger and reconstruction is of paramount importance: event topology, detector acceptance and response as well as particle reconstruction efficiency and precision must be understood in detail to avoid irretrievable losses and biases.

This was a new situation. Up to the 1970s, an experiment was designed using analytic calculations for the performance evaluation. Any shortcomings of the initial design would be “repaired” by appropriate analysis of the stored raw data: The prevailing mood was “you will sort it out later with software!” This new situation created a need, already at the design stage, for a complete simulation of the entire physics process of interest and of the underlying background: event generation, tracking of particles in the detector, interaction with the detector materials, signal generation and processing, digitization, many levels of trigger, pattern recognition, momentum or energy of all particles and finally comparison with the initial set of events [Highlight 9.5]. Besides being essential for detector design, simulation became indispensable for data analysis: simulated events, produced by reliable “event generators” and tracked with full detail through the detectors are compared with real data.

In the 1990s, projects were carried out in collaboration with industry to explore how computer architectures differing from the sequential von Neumann machine could exploit some form of fine-grain parallelism, but results were not convincing. The structure of the data, closely linked to the topology of the detector, consisting of independent elements, favoured a coarse grain parallelism, easily implemented by arrays of (sequential) commodity processors such as personal computers. In the

event selection, the first level trigger identifies a subset of data (ROI, Region Of Interest) where a detailed calculation should be performed and assigns this task to a single processor. Another ROI from a different event is assigned to another processor and the calculations proceed in parallel. When the result of an ROI points to a potentially interesting event, the corresponding full data set is assigned to a processor for further calculations. At the end of a successful selection the event is sent to permanent storage. The selectivity of the filter must ensure that the volume of stored data is manageable. This architecture has proved to be efficient and economical, and is used by LHC experiments [Highlight 8.11].

Off-line processing of data has followed a similar trend, moving from mainframes to event parallelism on clusters of workstations and PCs with efficient access to data storage.

In an experiment the bulk of the data comes from two sources:

- digitized signals from the detectors;
- data generated by simulation.

The raw data is first *reconstructed* to obtain a physics view of the event including elements such as particle tracks, computation of physical quantities, identification of particles, and classification of the event according to its physics characteristics. The output from this stage, the *event summary data* (ESD), forms the basis for further stages of analysis, including a comparison with corresponding ESDs generated by simulation.

The analysis is a distributed process involving many groups within the collaboration studying different physics processes, or the behaviour of the detector itself, all requiring access to the ESD. These groups will in general extract events of interest, perform additional processing, and create new datasets that are shared among the members of the group or within a small team.

There are several characteristics of the data processing which influence the design of the computing services:

- The volume of the data is large; currently, each year the LHC experiments generate about 30 Petabytes ( $30 \times 10^{15}$  bytes) of new data, raw, simulated and processed;
- The bulk of the data is shared between many groups, trawling simultaneously through large subsets of events;
- Only the raw (detector) data is static. Everything else is reprocessed as understanding of the physics, the detector, and the analysis programs evolves, and as the focus of physics interest changes;
- The larger data sets (raw, event summary, simulated) are read-only: updates are applied to *new versions* of the data set and the events are independent of

each other. Thus there is ample scope for parallel processing and replicas are relatively easy to manage.

The understanding and analysis of the data requires deep knowledge of both the physics and the detector. The active participation of specialists from many different areas is needed to adapt the reconstruction and simulation algorithms as new phenomena are studied. The large data sets are in no way self-explanatory: they are inseparable from the collaboration. As a result, data security is concerned more with protecting against accidental corruption or loss, rather than with encryption or restricting access.

The evolution of commodity processors in recent years, with many execution cores, an extended instruction set allowing operations on vectors and special purpose co-processors, requires re-thinking of the simple event parallelism. In the real time selection process, FPGAs (Field Programmable Gate Arrays) and video co-processors are used to prepare the data for the processor/core which performs pattern recognition, with large gains in event processing speed; some of the results of pre-processing replace raw data to permanent storage. In simulation many tracks can be handled simultaneously by using the vector capability of the processors.

Data analysis techniques have evolved with the complexity of detectors and the increased size of the data samples. In the bubble chamber time typically one had to extract a relatively narrow signal peaking on top of a uniform background, from a histogram representing a one-dimensional distribution. The signal was localized by “cuts” and its significance could be assessed by simple statistical methods.

In the LHC experiments the (rare) signals of interest are masked by a large and complex background arising from different processes. The search must be performed in a multi-dimensional parameter space, by choosing a set of efficient variables to enhance the signal-to-background, typically done by multivariate analysis (MVA). Different tools, such as neural networks or boosted decision trees, are used. Large sets of simulated signal and background events are needed to train the tools: Machine Learning (ML) becomes a step in the discovery process [9].

Significant progress has been made in recent years regarding the understanding of, and implementation of services and solutions for, long-term data preservation for future re-use [10]. Funding agencies and indeed the general public are now beginning to understand the need for preservation and sharing of data, which represent the “digital memory” of the experiments, as a part of the scientific legacy. This “memory” contains the raw data recorded by the experiments and also a significant amount of metadata, software programs and “knowledge”. The latter is tightly coupled to the collaboration and its members. The timescale for data preservation is determined by the availability of data from new experiments

exploring the same physics: as the lifetime of products in information technology is short, hardware and software obsolescence may have to be faced.

Data re-use is undoubtedly valuable for its discovery potential. However, it has an intrinsic limitation: the “raw data” kept for permanent storage are the result of the trigger process (taking place before the initial data recording) designed for specific physics reactions. The bulk of the data produced by the collisions are irreversibly lost and may carry with them undiscovered information.

Over the years, experiments have become larger and more complex, and the size of the collaborations reflects the size and ambitions of the experiments. Collaborating groups are located on all continents. This mode of operation is made possible by fast long distance networking [Highlight 9.4] and by software tools that ensure data coherence. It goes without saying that the Web [Highlight 9.7] and other means of digital communications have been essential to the success of these distributed collaborations.

## 9.2 Computing Clusters and Data Storage: The New Factory and Warehouse

Les Robertson

### *Infrastructure for innovation*

Since the earliest days of CERN the availability of computational and data storage capacity has been one of the most important factors in enabling the extraction of physics from the data collected by the experiments. It has evolved from the automated analysis of bubble chamber photographs in the 1960s to the worldwide grid used today to distribute the mass of data from the Large Hadron Collider for processing in 170 sites in 42 countries where high energy physicists have access to computing resources. Funding has always been limited, driving a continuous search for new technologies to provide more cost-effective solutions — in data storage, networking and processing power.

In the 1960s and early 1970s the clear choice for computational capacity was the super-computer, designed for fast floating point calculations, though not particularly good for handling large volumes of data. CERN’s innovative role in that era included working with the manufacturer to optimize mathematical libraries to exploit best the detailed architecture of the machine. The beginnings of the hierarchical mass storage management architecture were developed, the evolution of which is still continuing today. In the 1970s CERN already had a managed two-level data buffer (“cache”) for its vast magnetic tape library.

During these early years the rather visionary idea emerged that physicists needed more than just access to a large computer to run their programs. They

needed to acquire, store and analyse data from experiments and simulations, and do this from all parts of the CERN site. Experimental data files needed to be transported and processed on central batch systems, with control of the job flow and delivery of the results. With the first accelerators, data was transported on magnetic tapes to the computer centre “bicycle online” and physicists carried large trays of cards and reams of paper to and from their offices. But in 1968 the first of a series of CERN-developed site networks was implemented culminating ten years later in the high speed general purpose packet switched CERNET [Highlight 9.3].

By the mid-70s supercomputer architecture had evolved towards support for vectorizable codes. Sterling efforts to exploit this hardware for High Energy Physics (HEP) algorithms were unsuccessful and commercial mainframes with their simpler architecture became the cost-effective solution, ushering in a period of heterogeneity as the competitive acquisition criteria led over the years to the installation of systems from different manufacturers with their proprietary architectures, operating systems and local network media and protocols. The challenge was to interconnect these systems to the site network and, later, to the growing number of incompatible wide area networks that were appearing in support of science in different countries and regions.

By the end of the 1980s and the start-up of the LEP collider, CERN had acquired in-depth experience in storage management, networking and distributed processing. This was a key factor in three major developments that proved essential for HEP data processing over the following decades: cluster computing, extending this to the wide area as a data grid, and of course the Worldwide Web.

### ***Hierarchical mass storage service for the CDC 7600 supercomputer***

In March 1972 CERN installed serial number 3 of the fastest supercomputer of its day, the Control Data Corporation 7600. There were three large disks connected to the system providing about 2 Gigabytes (GB) of online storage. Another 600 Megabytes (MB) of disk storage was available on the “front-end” computers mainly used for holding data which was required to be permanently online such as executable files, but the physics data were stored on magnetic tape which had to be copied to the 7600 disks before being accessed. By the end of 1974 there were about 60,000 magnetic tapes in the tape vault, each with a nominal capacity of 20 to 40 MB [11]. In order to manage the movement of data between tape and the 7600 a two-tier storage system was developed in 1973. The user referred to data by the tape number and the system arranged for the tape to be recovered from the vault, mounted on a tape drive on one of the front-end computers and copied into a cache on the 7600 disks. The cache was managed using criteria such as size of file, age, and time since last access. A third tier was introduced a few years later as a tape reel cache, enabling frequently used tapes to be available for immediate



mounting. The system was called FIND [12] and at the time it was not described as a *Hierarchical Storage Management* (HSM) system, a term that was coined only later with the arrival of robotic tape handling devices, but it already had all of the characteristics of an HSM.

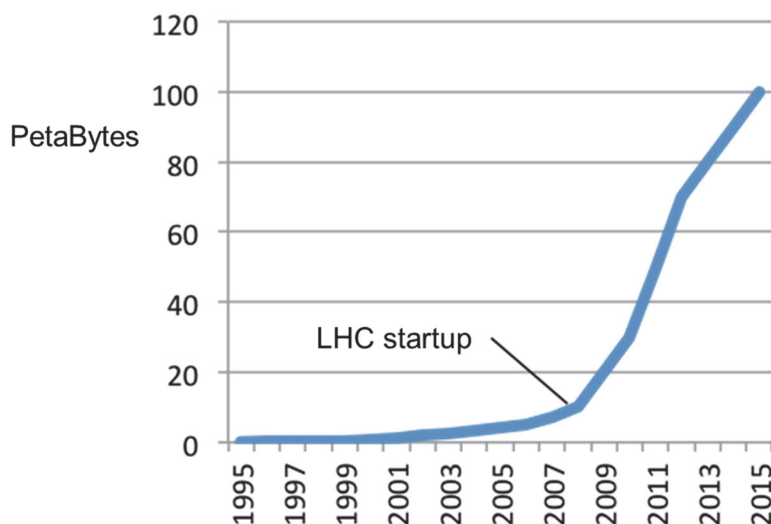


Fig. 9.2. Growth of mass storage usage at CERN.

The basic concept developed in 1973 has remained at the heart of storage management at CERN until the present day. There have been several generations of software integrating new technologies like robotic tape handlers and databases as the volume of data exploded (Fig. 9.2), and evolving to the cluster model with a fully distributed cache [13].

### ***BETEL — Extending cluster, processors and data, across the wide area***

In 1993 CERN led a project which implemented the first international demonstration of a computer cluster extended to the wide area using network technology with comparable performance to the local area networks of the time. The project, BETEL (Broadband Exchange over Trans-European Links) [14], used 34 Mbit/s ATM<sup>a</sup> (Asynchronous Transfer Mode) links provided by France Telecom and Telecom Switzerland, to interconnect the 100 Mbit/s FDDI (Fiber Distributed Data Interface) networks at CERN, EPFL (Lausanne, Switzerland), IN2P3 (Lyon, France) and EUROCOM (Nice, France).

---

<sup>a</sup>ATM was a standard used by telephone companies during the 1990s to provide high throughput data traffic combined with low latency characteristics suitable for voice and video. By the end of the decade it was being displaced.

Using this infrastructure the CERN physics cluster was extended to a similar cluster in Lyon, including access to the magnetic tape services at both sites. After enhancing the RFIO (Remote File Input Output) protocol [15] used for data access to take account of the long network latency, the performance goals were achieved enabling seamless access to all disk and tape resources from both sites. The project also demonstrated interactive physics analysis using CERN's Physics Analysis Workstation (PAW) system and a tele-teaching application used by users at EPFL and EUROCOM.

### ***SHIFT — An early implementation of cluster computing***

The computational needs of the experiments planned for the LEP collider from its start in 1989 far exceeded the capacity that could be provided within CERN's computing budget using the mainframe technology that was the backbone of the computer centre. In HEP data processing the major part of computational resources are used to process very large numbers of independent events. Events can therefore be processed in parallel, and using systems with different performance characteristics: the requirement is *high throughput* rather than *high performance*. A small team was given the job of looking for a distributed solution that could exploit low cost components integrated with the central computers and with the storage infrastructure essential for efficient data analysis.

There was good experience with microprocessors for online use of the experiments, but these had limited floating-point performance. Specialized processors had also been developed, implementing an instruction set sufficient for certain physics codes but without general I/O and network capability [16]. More promising were the "personal workstations" being used at CERN in a limited role for interactive graphics applications, and based on the new single-chip reduced instruction set (RISC<sup>b</sup>) processors. Personal workstations with the graphics functionality removed gave an order of magnitude improvement in price/performance over the mainframes. The only problem was how to integrate them into the physics analysis service.

The project SHIFT (*Scalable Heterogeneous Integrated Facility Testbed*) [17] developed a straightforward architecture providing scalability and accepting heterogeneity that would enable the service to expand smoothly as the underlying data storage and computational resources grew. It was seen as essential to construct the system with off-the-shelf hardware components, in order to be able to exploit new technologies and cost opportunities as soon as they arose. The key idea was to define separate services for mass storage, data cache management (disk) and

---

<sup>b</sup>Compared with traditional processors of that period RISC processors provided higher performance by using a simplified instruction set that enabled execution to be completed in fewer microprocessor cycles per instruction.

computation loosely integrated across a high performance network backbone. Each service would be implemented as a set of independent servers to ensure scalability. The software was designed to be portable across systems providing a Unix-like system interface and a TCP/IP<sup>c</sup> (Transmission Control Protocol/Internet Protocol) network service. The fundamental components that had to be developed were: a distributed cache management system integrated with the site mass storage service; a remote file access system [15], later, re-routing in the event of a link failure; a job scheduler to manage the workload across the cluster, initially based on a system developed by NASA.

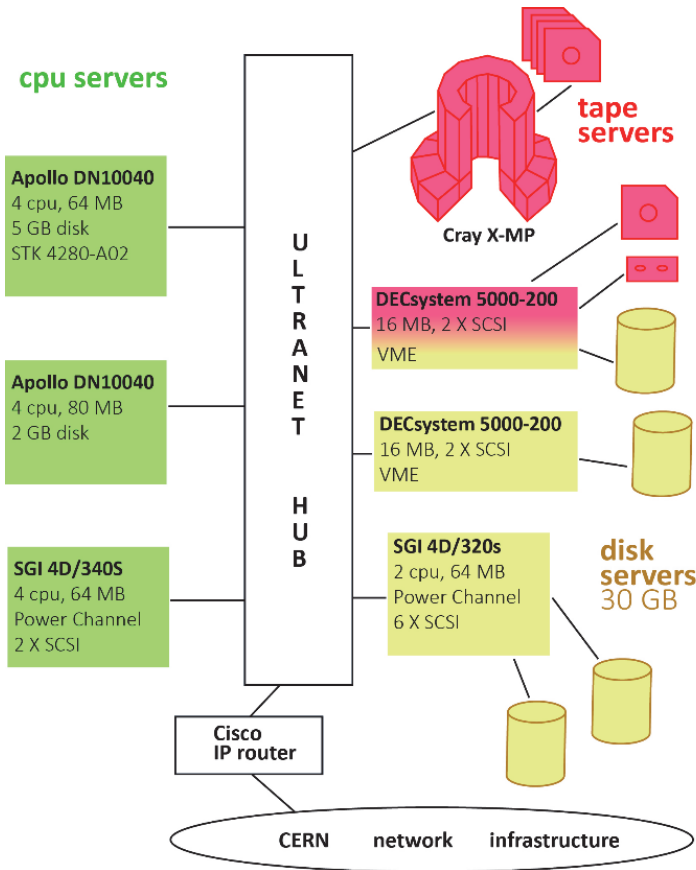


Fig. 9.3. Pilot SHIFT configuration at CERN in 1991.

<sup>c</sup>TCP/IP is the most widely used communications protocol. It was invented and developed in the 1970s at DARPA (Defense Advanced Research Projects Agency), USA.

The initial implementation of the architecture using Hewlett-Packard Apollo DN10.000 computers as disk and computer servers, could already rival the largest mainframe in terms of aggregate processing capacity, but lacked efficient access to mass storage and was only used for simulation work. The cluster was soon enhanced with the addition of systems from Silicon Graphics Inc., Digital Equipment Corporation, SUN Microsystems and a Cray X-MP as the mass storage server, all interconnected by a high-speed network from UltraNet. Figure 9.3 shows the configuration at the beginning of 1991. The unusual choice of a supercomputer as a mass storage server was made because, in common with the other systems in the cluster, it used a version of Unix as operating system, had a very good implementation of TCP/IP network protocols, and could be connected to the UltraNet backbone. From this early start, the cluster grew rapidly, and within a few years supplanted the mainframes as the workhorse of physics simulation and analysis. The architecture, in large part because of its simplicity, proved its flexibility, absorbing successive generations of disk and processor technologies. PCs were introduced in 1997 as soon as their performance on HEP codes became acceptable and they in turn soon displaced the RISC-based workstations.

### ***The Worldwide LHC Computing GRID — A Cluster of Clusters***

When planning the facilities for analysing the data from LHC began in 1998, once again it was clear that the budget for computing at CERN would fall far short of supplying the needs. The CERN cluster was by that time largely powered by PCs which seemed to offer a very promising future for raw performance and cost effectiveness — no better technology was on the horizon. However, many other HEP sites had also installed powerful clusters. A neat way had to be found of interconnecting them without the physicist having to worry about where the available resources were and where the data was located. A group of physicists and computing experts from a broad selection of HEP institutes got together to study possible solutions, producing a recommendation in 2000 for a multi-tier hierarchy of inter-connected regional centres, each with its own cluster and storage [18]. The tiers were defined by the type and volume of data that would be stored there: the top level, Tier-0 (CERN, and Budapest from 2013), would store a copy of all of the raw and simulated data and of the processed data required by all of the physicists in an experiment; Tier-1 sites would be large operations with mass storage services, and each Tier-1 would hold a copy of a fraction of the data held at CERN (ensuring at least two copies of all important data); lower tier sites would store only temporary copies of the main datasets, obtaining copies of datasets as required from a higher tier. The end-user would not have to know where the data was located or where there were free resources. The work would be automatically split into suitable processing units and directed to an appropriate regional centre.



Fig. 9.4. A grid of 170 interconnected computing clusters in 42 different countries provide the processing capacity for analysing LHC data (Sept. 2015). Budapest and CERN are Tier 0 sites.

While the responsibilities for operating services and managing data were defined as a *hierarchy* which would have implications for planning inter-site network bandwidth, it was important that the lower-level software should view the system as fully inter-connected, providing complete flexibility to job schedulers, data management services and other higher-level components: the logical architecture would be a *grid*. The *middleware* (software) required to operate the grid would be complex and rather than developing a special LHC solution it was decided to start with the *Computational Grid* technology being developed in the GLOBUS project in the USA [19]. The Italian Nuclear Physics Institute, INFN, already had experience with the Globus toolkit. The grid concept looked to be of general interest for scientific applications and CERN took the lead in launching an international multi-science project, the *European Data Grid* (EDG) [20], which received funding from the European Commission. EDG was a proof of concept project which developed an initial implementation of the middleware needed to complement the Globus toolkit and deployed this on a demonstration grid. A similar project, the Grid Physics Network (GriPhyN) was begun in the USA.

The grid concept and experience with early prototypes was promising and in September 2001 CERN set up the *LHC Computing Grid* (LCG) Project to move ahead with planning and deploying a grid on the scale required for LHC era. LCG collaborated closely with EDG, GriPhyN and successor grid projects, but with the clear goal of focusing on the functionality, delivery schedules, reliability metrics, interoperability and other factors needed to ensure an operational grid with the performance and timescales imposed by the LHC experiments. The first LCG

deployment was in September 2003 with 14 sites, followed by rapid growth in the number of sites and the installed resources. The differing goals of the R&D grid projects and LCG with its emphasis on service stability would be the cause of considerable conflict in the ensuing years, particularly concerning the functionality and timescales of the middleware. With the deadline of LHC start-up approaching, a compromise was worked out and the Worldwide LHC Computing Grid was in round-the-clock operation at 135 sites when data began to flow in 2008 [21].

A major concern in 2000 was the network bandwidth that would be available (and affordable) between the centres and much effort was invested in working with national and international bodies that provided network capacity for science to ensure that their solutions took account of the relatively high requirements of LHC. This work paid off and by the time that LHC began operation CERN was well connected to a large, mainly fibre, international infrastructure [Highlight 9.4]. Indeed, today, fifteen years after the first proposal for the distributed model, the available bandwidth enables a much more flexible model for data distribution: instead of scheduling work to where the data is located it is now possible in many cases to move data on demand to where there are free computational resources.

### 9.3 Local Area Networks: Organizing Interconnection

Ben Segal

The ever closer and increasingly important interconnection of particle experimentation and accelerators with Information Technology (IT) is the main theme of this chapter. In this highlight we illustrate one essential facet: the evolution of network concepts.

#### *Innovating in the Early Years — 1960s and 1970s*

CERN rarely develops computer networking technology *per se*, but is often an early adopter of the latest technology, operating it at its technical limits. This was different, however, during the 1960s and 1970s, when CERN had no choice but to innovate due to the limited offer of suitable commercial products. During these years several networking systems were developed in-house, presented here in chronological order, and reflecting the rapid evolution of IT and its growing importance for high energy physics. A comprehensive survey is given in [22].

In the early 1960s CERN's first computer, the Ferranti Mercury, was connected by a one-kilometre *data link* to electronic experiments in the PS experimental hall, making it the first computer at CERN to analyse on-line experimental data in "real-time". The complete link interfacing was designed and built in-house.

CERN's first distributed computer system (1968–1978) was FOCUS, which linked several experiments to the Control Data Corporation mainframes in the Computer Centre [23]. The link used 2 Mbit/s twisted pair serial lines via a smaller CDC machine serving as a concentrator and front-end. This allowed physicists at their remote experiment sites to manage their files, to execute batch jobs on the mainframes, and to retrieve their results.

The next step was the OMNET network, CERN's first packet switched network (1971 to the early 80s) [24]. It eventually interconnected around 30 user minicomputers as well as the central IBM and CDC mainframes, via dedicated switching nodes. A small purpose-built CERN operating system, SMO, in a high-level language, PL-11, had to be developed for these nodes. The network protocol software, and data link hardware and interfaces were developed and built at CERN.

From 1971, twelve Remote Input Output Stations (RIOS) allowed remote batch job submission to the central CDC mainframes from remote CERN sites. Users no longer needed to transport and physically input cards and output printouts to and from the Computer Centre — a great gain in efficiency. Each minicomputer-based RIOS emulated a standard CDC remote batch terminal, but was much cheaper and more performant. Through one particularly important innovation, the human operator, who had been needed to initiate and supervise a computing job, was emulated (and replaced) by RIOS software. This strategy was successful and further expanded [25].

A technological prescient development was CERNET, a packet switched network, designed from 1975 and deployed from 1977 to 1988 [26]. CERNET offered a “Line Protocol” between switched interfaces, capable of up to 8.5 Mbit/s over short parallel connections, or serial line speeds of around 2.5 Mbit/s over twisted-pair cabling of up to 2 km (with repeaters). An “End-to-End Protocol” connected host to host applications. Once fully deployed, CERNET interconnected the central IBM and Control Data mainframes and over 100 minicomputers. The main CERNET application was the “File Access Protocol”, but a “Virtual Terminal Protocol” (remote logon) accessed the main IBM system.

Not to be outdone by the experiments the accelerator builders used IT technology in innovative ways too. For example, *Accelerator Control Networks* were designed and built to control the SPS. This was based on a star topology, with a message handling computer at the centre of the star [Highlight 5.2].

## ***Profiting from the IT Revolution: 1980s and 1990s***

Starting in the 1980s, the computing and networking world was shaken by several revolutions. CERN seized this opportunity, following these developments and collaborating closely with industry in a bid to anticipate the optimal choices for hardware and software, as outlined below.

### **Choice of TCP/IP and Internet**

Throughout the 1970s and 1980s there was a bitter contest between “US” (TCP/IP Internet) protocols and “European” (ISO-OSI) network protocols, as well as between diverse commercial network standards. Despite being in Europe and despite much pressure, CERN decided to test TCP/IP. Starting in 1984, the installation of TCP/IP on most of CERN’s computing platforms was as innovative as it was audacious. By the end of the 1980s TCP/IP and the Internet had clearly become dominant and CERN could benefit fully from its courageous choice. CERN joined the global Internet in 1989 [27]. The later Web and Grid developments at CERN all depended vitally on this work [Highlights 9.2 and 9.7].

### **Choice of Ethernet**

The first widely used Local Area Network (LAN) was the 12 Mbit/s token ring network supporting the Apollo Domain and its powerful workstations, starting in 1982. It used thin coaxial TV cable, requiring dedicated installations. Ethernet (10 Mbit/s) arrived in 1983, using different types of coaxial cables, followed by a token passing ring at 4 Mbit/s for LEP controls in 1986 [28] and much later by FDDI at 100 Mbit/s on optical fibre cables. Fortunately, Ethernet was soon recognized as the future dominant standard.

### **CERN’s central Ethernet services**

A centrally-managed Ethernet service was first set up in the mid-1980s on coaxial cables at 10 Mbit/s. Early difficulties due to the multiple network protocols present at that time had to be overcome, but the rapid evolution of Ethernet technology allowed CERN to homogenize its network infrastructure. The media (and prices!) improved dramatically and network speeds passed from 10 Mbit/s to 100 Mbit/s, and finally to Gigabit Ethernet. Today, Ethernet networking is simply a matter of choosing and installing commercial products. LHC controls are based on Gigabit Ethernet, running mostly via optical fibre [29], and distributing machine synchronization signals with sub-nanosecond accuracy [30].

### **Choice of Unix/Linux**

Unix first appeared at CERN in 1981 but was hardly noticed in spite of its important role in connecting CERN, and later all of Switzerland, to the worldwide email and news network USENET. It was widely mistrusted compared to



proprietary operating systems. Unix really came out of the closet when chosen for the Cray XMP installation in 1987. CERN and Cray worked together to develop the new UNICOS system, the first to support a full-fledged mainframe, and later to be the model for the SHIFT “distributed mainframe”. Unix was chosen as the operating system for LEP controls in 1988.

While Unix had proved itself at CERN by the late 1980s, it was still for the most part a commercially supported product. The open source version, Linux, supported by the online programmer community, was mistrusted, raising fears that “we will have to fix its bugs ourselves”. When finally accepted, during the final migration of SHIFT to PC/Linux platforms in the late 1990s, it rapidly became dominant.

### **Choice of the “SHIFT” architecture**

It was a game- and life-changing choice to use commodity CPU, storage and network hardware to build a “distributed mainframe” in a scalable, flexible and economic way, starting in 1990. CERN’s “Scalable Heterogeneous Integrated FaciliTy”, SHIFT, was the precursor of today’s cluster and cloud computing systems which completely replaced mainframes.

### **The networking issues for SHIFT**

SHIFT’s CPU, Disk and Tape server nodes had to be linked by a “Network Backplane” with, for the time, extreme throughput requirements (10 Mbit/s for a small system). Early on, it also had to deal with the high CPU power consumed by network I/O at the nodes. This was solved by the UltraNet [31] system. Later the backplane became a hybrid system containing elements of Ethernet (10 Mbit/s), FDDI (100 Mbit/s), UltraNet (600 Mbit/s) and HiPPI (800 Mbit/s) [32], until switched Gigabit Ethernet became sufficient for the task. CERN contributed to the further development, which led to the faster 6400 Mbit/s Gigabyte System Network (GSN) [33], a precursor of InfiniBand [34] now used for the data acquisition of the LHC experiments.

## **9.4 High-Speed Worldwide Networking: Accelerating Protocols**

François Fluckiger

Public research networking created the Internet. Rarely in history did the academic community have such a dramatic and profound impact on the economy, and on society. As part of this community, the role of CERN was essential. CERN is well-known for its invention of the Web, but less so for its contribution to Internet infrastructure, central to the development of the worldwide Internet, and for its leading role in the development of high bit-rate networking.

### ***Paving the way to the Internet and Web***

In the early 1980s, CERN's external connections amounted to two minuscule leased lines operated at 9.6 kilobits per second (kbit/s), one to the Rutherford Appleton Laboratory (RAL), the other to CEA, Saclay. Ten years later, CERN had become the centre of a large star-shaped network, by far the largest internet hub in Europe. Today the LHC Optical Private Network (LHCOPN), a network connecting CERN to the LHC T1 sites operates multiple 100 Gigabit per second (Gbit/s) links. How did CERN get there?

Facing technological challenges has frequently been at the root of developing leading-edge technologies for CERN engineers. In turn, mastering a technology provides the confidence necessary to undertake very ambitious projects. Such was the case for networking, where CERN developed as early as the 1970s outstanding expertise on network protocols.

Facing the need for an open (i.e. based on non-proprietary technology), high bit-rate communication system to interconnect on-site a variety of computer brands — mainly in order to transfer files — CERN launched CERNET [26], the most ambitious networking project of its time. Communication between computers requires the use of common rules. These rules are called communication protocols. They are building blocks stacked one on top of the other in a layered structure, and are in practice implemented as modules of hardware, software or a mix of the two. They broadly divide into two types. The first type ensures the transport of raw information data between two network points, independent of its significance or use. The second type of protocol offers the end user a tangible service, e.g. email. CERNET implemented a genuine and complete suite of layered protocols, ranging from the physical and electrical plug to the CERN File Transfer Protocol running on half a dozen different brands of computers. CERNET was put into operation in 1977. This is a unique example of a complete multi-brand networking stack developed by a single organization. It was also the fastest packet-switching network of its time [Highlight 9.3].

Probably more important for the future was the type of technology adopted by CERN for one of the layers, called the packet layer. All modern networks use the packet principle where the data stream is chopped into individual pieces called packets. Packet protocols are divided into two fundamentally opposite classes: the connection-oriented class where the two end-computers must establish a connection before an information packet can be sent — the equivalent of a telephone call, and the connectionless class, where packets are independent of each other, as letters posted in a letter-box, and can be sent at any time. Connection-oriented networks check first via the call set-up mechanism that a fixed end-to-end route is available. The main advantage lies in the fact that commercial network

operators know the duration and the amount of traffic exchanged over each connection and they can charge the user on a per-call basis. The drawbacks are the extra delay to setup the call before any useful data can be sent and the additional complexity. Connectionless networks are not aware that a series of packets may belong to a flow. As letters, individual packets carry the full destination address. The drawback is that no prior check is made that a non-congested route is available when new packets are admitted. The chief advantages are the no-delay admittance of packets, higher resilience as routes are not pre-assigned, and more generally, simplicity.

CERN opted in the early 1970s for the latter, the connectionless approach. A prescient decision, since connectionless is precisely the philosophy of the Internet Protocol, the famous IP. By creating, 15 years before the explosion of IP, a deep expertise in IP-like technology, CERN paved the way for its future role in the Internet infrastructure and also for its invention of the Web.

### ***The emergence of HEPNet***

In the early 1980s, only RAL and Saclay were connected to CERN, via slow analogue leased lines funded by these two organizations. They were mainly used for remote login to CERN mainframes. In 1983, a proposal was made by CERN to connect more sites and offer additional services (File Transfer, Job submission and email in addition to remote login). This initiative was coined HEPnet. As alternative to leased lines the use of the emerging public packet switched networks was proposed by the national PTTs, monopolies at that time. The service offered by the PTTs was based on X.25 technology, a connection-oriented protocol, which is the exact opposite of CERNET and IP. CERN connected to the Swiss PTT X.25 network, allowing computers to make calls — as over a telephone network — to and from any other HEP site connected to their domestic X.25 network. To facilitate file transfers, CERN developed a gateway between CERNET and X.25, which was released in 1983. This development was one of the very few operational gateways between connectionless and connection-oriented networks.

The main drawback of X.25 networks was their per-call charging principle modelled by the PTTs on the telephone, making difficult any planning of the telecommunication costs. The X.25 PTT service was progressively abandoned and new leased lines — initially analogue, then digital at 64 kbit/s — connected additional HEP sites to CERN. In parallel, in 1982 IBM proposed to several research computer centres in Europe, to establish and fund for four years a network based on the BITNET (Because It's Time NETWORK) model in the USA. The initiative was called EARN (European Academic and Research Network) [35]. Additional analogue links were installed and CERN became in 1983 the largest hub of EARN. At that time, all of these links operated in a proprietary way.

### ***CERN: Precursor in Internet Technology***

In 1985, a TCP/IP coordinator was appointed at CERN and the LEP control system adopted this technology. CERN was one of the first sites in the world to receive for evaluation early versions of IP routers from the future market leader. It was at the end of 1988 that CERN started to operate leased lines with the TCP/IP protocols. By then, more leased lines had been set up, extending to Boston and Tel-Aviv. In summer 1988, a CERN team went to Beijing to organize the very first academic connection between China and Western Europe. But more than the number of connections, it was the tremendous increase of the bandwidth that changed dramatically the resulting services. In 1988, most analogue links had been replaced by digital circuits, at multiples of 64 kbit/s. Furthermore, by funding the first European 2 Mbit/s academic link in Europe, between CNAF Bologna and CERN, INFN propelled external networking into the Megabit per second era. No ground links of that bit-rate were available, but experimental satellite channels were. From 1981 to 1983, the STELLA project evaluated protocol aspects of 2 Mbit/s satellite channels between CERN, Pisa and RAL.

### ***CERN becomes the largest Network Hub in Europe***

In 1990, the very first Internet transatlantic megabit link (1.5 Mbit/s) was set up between CERN and Cornell University. A full TCP/IP connection to the US National Science Foundation Network (NSFNet) was established, the link being the result of EASYNet, a new IBM initiative in Europe, three years after the end of their EARN funding.

In 1991, the star network around CERN became the largest internet hub in Europe: 80% of the Internet capacity in Europe for international traffic was installed at CERN (Fig. 9.5). It is no surprise that the performance of Tim Berners-Lee's first Web server impressed the world: it was at the heart of the European Internet, just a few hops from most destinations. Why did this occur? There was an absolute need for high bit-rate access to CERN from HEP centres so that physicists could work remotely almost as if they were on-site at CERN. In addition, networks grew like crystals: as one hub started "crystallizing", everyone wanted to connect to it being inspired by the novel opportunities offered. Therefore, non-HEP scientific centres started to set up links to CERN, often serving as relays to other domestic HEP sites. Finally, a networking hub, being a central point of failure, requires extremely reliable operation. Inherited from the accelerator culture, CERN had a tradition of excellence in 24 hours/7 days operation that was recognized worldwide.

### ***CERN initiates IP address allocation in Europe***

Another key contribution of CERN to the Internet infrastructure was the attribution of IP addresses. Any device connected to the Internet needs a unique identifier: the famous IP address. Before 1989, all addresses were allocated by the Internet Assigned Numbers Authority (IANA) in the USA. It was CERN, which convened at its site in December 1988 the first meeting to discuss the creation of a structure in Europe to allocate the European addresses. Six months later, the first meeting of RIPE (Réseaux IP Européens) took place in Amsterdam. RIPE is still, 25 years later, the authority that allocates Internet resources and services in Europe, in the Middle East and parts of Central Asia [36].

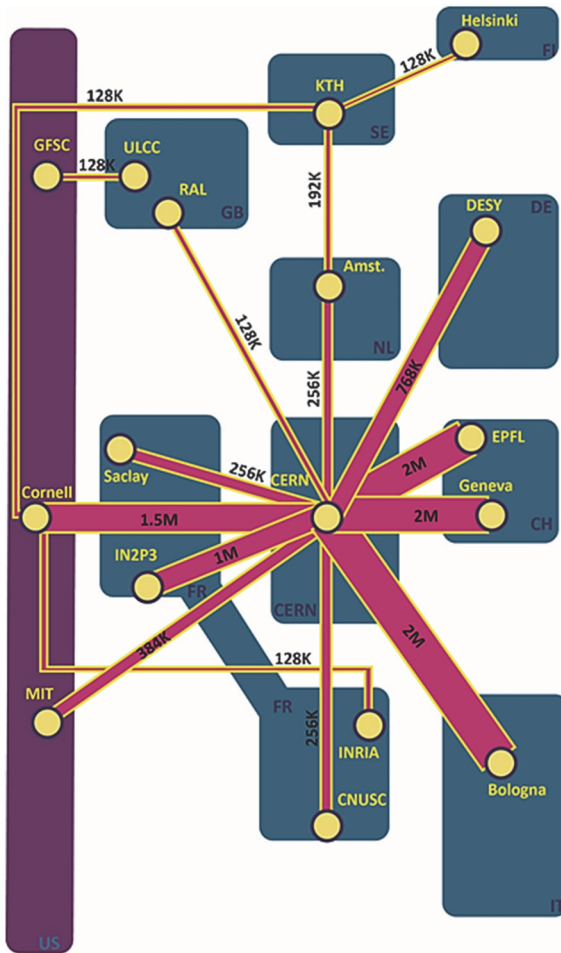


Fig. 9.5. The star network around CERN in 1991. Only high throughput data links are shown: for all links see <http://cds.cern.ch/record/2054391>.

Having reached its zenith in 1991, the CERN share in the Internet infrastructure slowly decreased, for two reasons. First, though prepared to remain a major Internet Hub, CERN considered that it would be healthier for the Internet if other hubs emerged. Second, in many countries, National Research and Education Networks (NRENs) had emerged, to which most HEP centres were connected [37]. These NRENs were, and still are, interconnected via the pan-European academic backbones operated by the DANTE organization, initiated by the European Commission [38]. The connection of CERN to these successive backbones moved from 34 Mbit/s in 1997 to 10 Gbit/s in 2005. In that period, CERN played a similar role to that of the European NRENs. However, there was still a need to push the technological limits and in 2003, in the context of the EU-sponsored CERN DataTAG project, the world record for data transfer was broken by CERN, together with the California Institute of Technology (Caltech), the Los Alamos National Laboratory (LANL) and SLAC, with the successful transfer of 1 terabyte of data in less than an hour over the 10,037 km between CERN and California.

### ***The Gigabit era***

To meet the requirements of the Worldwide LHC Computing Grid (WLCG) [Highlight 9.2], CERN and its HEP partners returned to the approach of HEP-dedicated links between the CERN and the major HEP centres (Fig. 9.6).

These dedicated ultra-high bit-rate links are provided by GÉANT [37], a pan-European research and education network supported by the EU, and by other NRENs: they form the LHCOPN (LHC Optical Private Network). The minimum bit-rate is 10 Gbit/s and several circuits operate at higher speed including a dual  $2 \times 100$  Gbit/s transatlantic link to the Energy Science Network (ESNet) in the US.

Through the deployment of the Internet infrastructure technology, CERN, together with the rest of the academic community, has also had an indirect impact. Academic organizations — be they universities or research centres — are sometimes compared to industrial companies in unfavourable terms regarding aspects such as efficiency, organization and rigour. This is perhaps viewed as a price to pay for creativity — a perception inherited from the old idea that inventiveness and organization are antagonistic qualities. The academic community was sometimes viewed as being only able to invent new technologies, but not to contribute to their development. The development of the academic and research networks has demonstrated that the academic community is capable of deploying globally extremely complex systems. They are capable of delivering professionally operated systems, which can revolutionize the economy in a few years and permeate all levels of society.

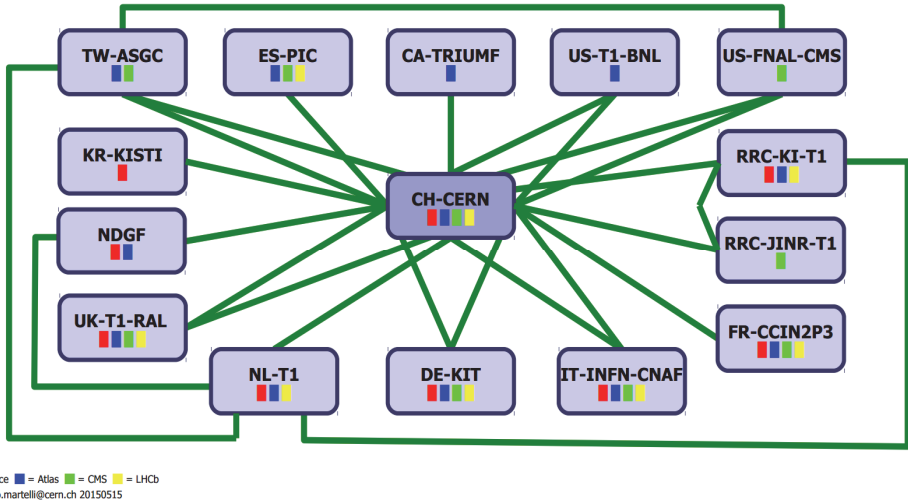


Fig. 9.6. LHCOPN topology. Details at <http://cds.cern.ch/record/1997396?ln=en>.

## 9.5 Detector Simulation: Events Before the Event

René Brun

Simulation<sup>d</sup> of a particle physics experiment relies on a chain of models which follow the evolution of the particles produced in the interaction up to the signals generated in the detectors. These can be:

- Models of the physical process representing the primary goal of the experiment and of all the known processes that could be considered as background;
- A model of the detector (geometry, materials, magnetic and electric fields) and of the path and interactions of each particle through the detector elements;
- Models of the signals generated by the particles in each detector type, followed by the electronic processing to produce digital information;
- A model of the selection procedures (trigger) which retain the wanted topology and reject background.

All models in the various steps are complex and contain many variables: simulation of an experiment is possible only by using Monte Carlo methods.

In the end, the simulation chain produces “data” in the same format as the real experiment. Models of the physical processes are needed to test the viability of the initial idea. Geometry and acceptance go in step with the overall design of the

<sup>d</sup>A simulation uses a mathematical description, or model, of a real system in the form of a computer program (Encyclopaedia Britannica).

apparatus. Signal generation is useful to understand in detail the behaviour of individual detectors. Simulated signals are used to train and tune selection and pattern recognition procedures. Simulated raw “data” help testing the reconstruction software.

Once real data become available and are processed, results are compared with the output of the simulation chain. In order to extract efficiently a potential signal from the background, one must identify the most efficient (uncorrelated) variables. Multivariate analysis (MVA) treats many statistical variables simultaneously, taking into account the effects of all variables on the response of interest. The tools available, such as neural nets, “boosted” decision trees, etc., go under the name of Machine Learning (ML). These tools make extensive use of simulated “data” to train their decision algorithms. TMVA [39] (Toolkit for Multivariate Data Analysis with ROOT) is used by the LHC experiments for this task.

As the understanding of the experiment improves, more refined simulations may be desirable, requiring running the whole cycle several times.

The full simulation cycle is very demanding of computer resources. Different methods of “fast” simulation have been devised to replace the full simulation of sub-systems by introducing simplified assumptions on detector geometry or by skipping some Monte Carlo calculations and using the results from formulae or from a stored catalogue. Computing time can thus be reduced by a large factor.

Fast simulation is not a replacement of full detector simulation. It must be tuned/validated with full simulation results (while full simulation is itself tuned/validated with data containing well-known reference signals). Fast simulation is used primarily for quick and approximate estimates of signal and background rates. It is also useful for an initial survey of complex, multi-dimensional parameter spaces which require a very large (simulated) data set.

Simulation is essential in designing experiments and interpreting the results, not only in high energy physics (HEP), but also in many other branches of science and technology. Simulating the known phenomena with good precision is a necessary prerequisite to the discovery of new effects, by comparing simulation with data.

Event generators have been developed for simulating interactions of different primary particles and a wide range of energies. An interaction process is described mathematically by a “matrix element” which must be integrated over the multidimensional internal parameters phase space to provide the total and differential cross-sections. Each point of this phase space is associated with a probability. According to this probability events are randomly generated.



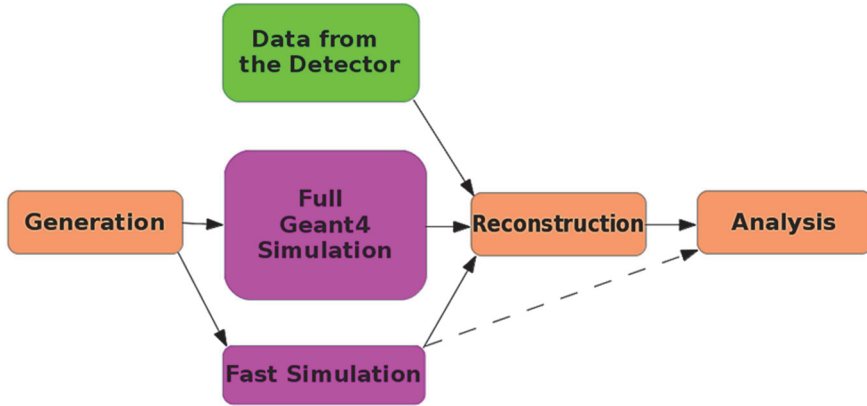


Fig. 9.7. Simulation in the LHC experiments.

As an example, consider the steps involved in the generation of events in proton-proton interactions as they happen at the LHC. The initial state consists of colliding protons, but the most interesting interactions are the hard collisions between constituent partons [Boxes 6.4 and 8.3]: the simulation must first take into account the distribution of partons inside a proton, as described by its structure functions. Then the specific “hard interaction” between partons has to be selected. As the most interesting processes (e.g. Higgs physics) are very rare compared to the dominant QCD background, the latter must be correctly simulated, which implies considering many relevant loop diagrams [Boxes 4.2 and 5.1]. Other partons belonging to the initial protons may interact “softly”, and must be also simulated as they contribute to the complexity of the event. Following the initial hard interaction, the bare partons do not exist in nature (they are confined inside the hadrons) and must be “dressed”, to form the known hadrons or mesons. This goes in steps: parton shower (describing their radiation), hadronization (grouping partons into particles) and decay of unstable particles. The task of comparing simulated events with data is further complicated by the fact that at the LHC several proton-proton interactions take place at each beam crossing.

Event generators must consider the physics complexity at the appropriate level together with the most efficient simulation methods. The goal is to generate enough simulated events to match the number of real events with affordable computing resources. An event generation package widely used for high energy collisions of electrons, positrons, protons and antiprotons is PYTHIA [40]; other packages have been developed for specific processes or initial states (e.g. heavy ion collisions).

Up to 1975 the major software development for HEP was associated with bubble chamber experiments. Event kinematics was reconstructed from the

information obtained by scanning and measuring pictures. With early electronics-based experiments, the reconstruction software was tested and run only after data taking. Events consisted of straight or curved tracks hitting detector “planes” (digitized spark chambers or multiwire proportional chambers) and showers in calorimeter “blocks” (sodium iodide or lead-glass). EGS [41] from SLAC was the state-of-the-art program for simulating an electromagnetic shower in the simple geometry of these calorimeters. EGS users had to implement essentially two functions: to tell the system “where I am” or “where do I go”. With the growing complexity of the detectors one soon realized that full simulation (tracking + calorimetry) was essential for testing the reconstruction programs and computing the geometrical acceptance. In 1975 a combined effort between the Omega, NA3 and NA4 experiments at the SPS led to the Geant1 and later Geant2 simulation tools (Generation of Events ANd Tracks). These provided a simple environment to propagate particles produced by an event generator, specific to each experiment, through the various detector modules, taking into account magnetic fields and multiple scattering. In 1981, a major upgrade to Geant3 [42] provided a tool to describe geometry with an automatic transport mechanism. Geant3 was interfaced to EGS3, then EGS4 and several hadronic shower packages. Geant3 included major developments and became the main simulation tool for the LEP detectors, for experiments at the Tevatron and RHIC and for the design of LHC detectors.

In 1994, the Geant4 [43] system was implemented in C++, based on the same principles as Geant3 as concerns detector description and transport. Today it is the cornerstone for all detector simulations in the HEP world. During phase1 of the LHC, about 55% of the total CPU time on the GRID machines was dedicated to detector simulation. It has become essential to improve the performance of the simulation tools, and a new project GeantV was launched in 2012 with as objective to gain possibly a factor 10 in speed. Both Geant3 and Geant4 had and still have a strong impact outside the HEP world, contributing to the design and analysis of many devices in astrophysics, medicine, hadron therapy, materials science.

Another development should be mentioned in the field of simulation: FLUKA [44]. Started as a program to calculate radiation shielding at the PS in the early 1960s, it was refined for the same purpose to cope with the higher energies of the SPS. It went through a major upgrade which made it a general purpose tool for calculations of particle transport and interactions with matter, covering an extended range of applications from proton and electron accelerator shielding to target design, calorimetry, activation, dosimetry, detector design, cosmic rays, neutrino physics and radiotherapy.

## 9.6 Data Analysis and Programming Environment: Distilling Information

René Brun

### *Data storage formats and analysis tools*

Data from bubble chambers and optical spark chambers were recorded on photographic film. Points on selected tracks were measured on the film and stored on magnetic tape as “raw data” in relatively simple formats specific to each experiment. Raw data from the early experiments using digitized spark chambers and simple calorimeters were also stored in experiment-specific formats. Then the reconstruction program was processing these events and writing a “Data Summary Tape” (DST) with compact data types for the reconstructed tracks (direction, momentum) or calorimeter information. The next step was data analysis: The task was to assign or calculate particle masses and other quantities of interest, store them and display selected information in histograms. During the 1960s, the operation was optimized for the bubble chamber experiments by producing a suite of programs THRESH, GRIND, and SLICE, implementing the three stages of geometry, kinematics and assignment of mass to each particle: they could be used by all experiments with little customizing [45]. During the early 1980s, it was realized that this process could be simplified by storing the quantities of interest in “n-tuples” (a simple row of  $n$  variables put into tables). The Physics Analysis Workstation PAW [46] system developed in 1983 was based on row-wise  $n$ -tuples. A user could interactively process the  $n$ -tuples data with simple queries. Histograms were automatically viewed in many possible graphics formats on the user workstation. The system became rapidly very popular for data analysis and  $n$ -tuples grew larger. PAW  $n$ -tuples became the main analysis format for most experiments at CERN and other labs between 1988 and 1998.

In 1995, the ROOT project was started based on the same philosophy as PAW, but implemented in C++ and able to support more complex data types [47]. By 2000, ROOT had a built-in system capable of accepting most C++ constructs found in the LHC experiments. ROOT had been selected in 1998 for the Tevatron experiments at Fermilab, followed in 1999 by the RHIC experiments at BNL. All LHC experiments, as well as most HEP or related laboratories in the world adopted the ROOT system. ROOT was progressively interfaced with many other tools (graphics, user interfaces, statistics, mathematical libraries, etc.), becoming de-facto the new CERN Program Library. About 20,000 people download the ROOT binary or source files every month and about 9,000 people are registered to the ROOT Forum. Besides being used by more and more people in the scientific domain, ROOT is also used in the car and oil industry, and in finance.

### ***Programming Languages***

For about four decades, FORTRAN was the main programming language. Simple to learn and well adapted to small or medium size single process and sequential applications, it was the language of CERNLIB, the CERN Program Library [48], built from contributions of users developing simple algorithms, linear algebra or general utilities with up to a few hundred lines of code. CERNLIB was very popular in the scientific community: during the 1970s, it included task-oriented packages with a few thousand lines of code, e.g. GD3 for basic graphics, HBOOK [49] for histogramming, MINUIT [50] for fitting and minimization. At the same time data structure management systems were developed. These contained both numerical information and logical information about the object they describe, and included HYDRA [45] for bubble chamber experiments and ZBOOK [51] for electronic experiments. The ZEBRA system (ZBOOK+HYDRA) appeared in 1982, enabling the creation and management of a complex network of dynamic data structures, and saving them to files in machine independent format.

With the advent of FORTRAN90, it would have been possible to upgrade ZEBRA, taking advantage of the new features (similar to C structures), but this did not happen: FORTRAN was less and less competitive with the new software programming systems that provided user interfaces, graphics, network communications and interfaces with the operating system and hardware devices. With the increasing number of personal workstations and network connections, more and more applications required calls to the operating system, file and network system. These interfaces were typically implemented in the C language. For example, in the PAW system a substantial fraction of the user interface and graphics was in C.

The years 1990–1994 witnessed many discussions about the future language for HEP. By the end of 1994, it was agreed that the programming language for HEP was going to be C++. Two Research and Development projects were launched: RD44 for the development of Geant4 and RD45 for the evaluation of Object-Oriented data bases. Starting in 1995 the ROOT system also opted for C++.

### ***Programming Environment***

The size of programs has grown with time at a tremendous rate. In the early 1970s, large programs were around 10,000 lines of code. CERNLIB was some 30,000 lines of FORTRAN and assembler in 1970. The HBOOK or HYDRA packages contained around 15,000 lines in 1975. CERNLIB reached a maximum of about 150,000 lines in 1985, including the major packages. ZEBRA stabilized with about 30,000 lines in 1985. Geant3 started with ~10,000 lines in 1981 and reached ~200,000 lines in 1993; PAW reached a maximum in 1994 of ~250,000 lines. During the 1970s, an experiment-specific code (essentially the reconstruction

program) was less than 10,000 lines, reaching about 100,000 lines at the start of LEP in 1989. Today, Geant4 contains ~1.5 million lines, ROOT ~3 million lines, with specific software on top of these packages weighing in at ~5 million lines per experiment — not counting the analysis software by thousands of physicists. The memory used by these programs was around a few tens of kilobytes in the 1970s. The full simulation of the OPAL detector planned for LEP in 1981, including electromagnetic and hadronic showers, could run on central computers having only 1 MB of memory. From LEP to LHC experiments have grown in size and complexity, and the signal to background ratio has become much more demanding: today one sees simulation systems requiring 3000 times more memory! (Fig. 9.8)

Managing the source code of HEP software has always been a concern, even when systems had only a few thousand lines of code, because the software development has always been a joint effort in experiments. The PATCHY system [52] was developed at the end of the 1960s for bubble chamber software. A PAM file (PATCHY Master file) was common and read-only for all developers. Each developer prepared his/her own file with changes or additions to the master file. A PAM file was updated periodically by user consensus. PATCHY remained the preferred solution in the 1980s until an interactive version of PATCHY, called CMZ, appeared in 1989, making it easier to develop software on distributed systems by using workstation with nice text editors. PATCHY/CMZ were replaced

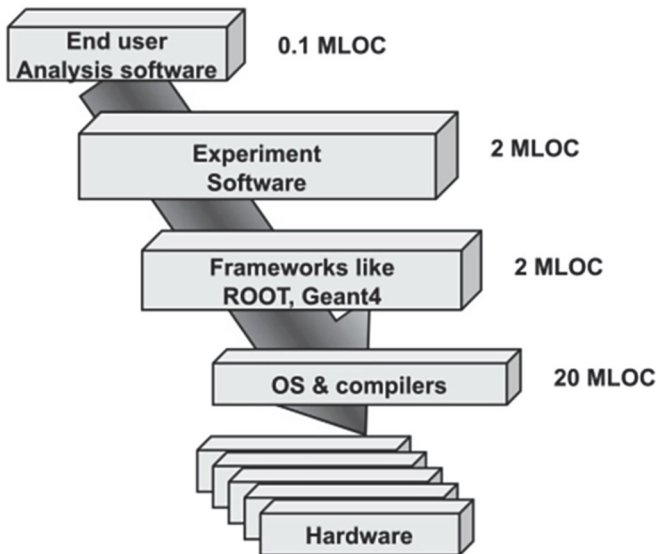


Fig. 9.8. The Software hierarchy (MLOC ↔ Million Lines Of Code).

by the open source CVS (Concurrent Versioning System) appearing at the end of the 1990s. With CVS, it became possible to edit/manage a central code repository shared by hundred/thousand developers, showing differences between successive versions and making easier code releases. Around 2005 an evolution of CVS, called SVN (Sub Version System) was adopted for all developments in LHC computing (and everywhere else). The GIT system replaced SVN around 2012, offering improved capability to manage successive versions of code developed and used by thousands of people. This trend will likely continue, following the continuous requirements to manage many millions of lines of code.

Error reporting systems have also evolved in the wake of the growth of the software size. From pure oral reports in coffee meetings or paper mail, huge progress followed the installation of the local area CERNET, followed by BITNET in the early 1980s, which brought worldwide e-mail exchange. The first web-based systems appeared around 1995 allowing the visualization of the source code and its documentation. The first common error reporting and managing system SAVANNAH appeared in 2000, followed by the current JIRA system in 2012. Several projects also host a discussion forum where users can report questions and problems, so that more experienced users can help (e.g. the ROOT forum).

### **Outlook**

Computing at CERN has followed and continues to follow the general trends in the field. The computing languages C++, Python, Java are now the main languages in science, industry, communication and information. This approach facilitates communication among these various fields, but is also an important asset for the careers of young scientists who seek employment outside High Energy Physics (HEP).

General tools like ROOT will continue to evolve following the general trends not specific to HEP like user interfaces, graphics and networking. A considerable effort is devoted to integrate machine learning libraries and techniques to facilitate the ever more complex data analysis phases. The Open Data philosophy facilitating the use of experiments data for teaching will also require simple and common user interfaces, professional documentation and data descriptions.

The detector simulation with the Geant family will have to evolve to cope with the high luminosity machines and increasing energies. Factors of 10 to 100 in processing speed will have to be gained, taking advantage of parallel architectures and by integrating fast simulation algorithms into the full and therefore necessarily slow simulation system. Machine learning techniques will enter this field to optimize automatically the simulation and analysis processes.

## 9.7 World Wide Web: Global Networking

François Fluckiger

The World Wide Web was born at CERN. As is the case with many births, the coming of the Web was a combination of chance and necessity.

In June 1980, a team of half a dozen “rent-a-programmers” were hired by CERN from a British electronics company to work for a few months on the PS Control System. Luckily, in addition to writing software, they had been asked to thoroughly document their work. This was a complex task as most modules were inter-dependant. One of the young engineers proposed to use a special technique to reflect the dependencies and references between the multiple parts of his own documentation. The small system he wrote was called Enquire. When he left CERN at the end of the contract, the floppy disk with the code of the Enquire system ended up in a drawer and no-one ever reused it [53]. The technique that had been prototyped for the Enquire system was *hypertext* and the name of the young engineer was Tim Berners-Lee. The hypertext concept was not invented by him, and Enquire was not its first implementation. The principle had been laid down in 1945 and the term was coined in 1963 in the context of the Xanadu project [54], followed by the first demonstration of a hypertext system in 1968. Yet, in 1980 two things had happened: the hypertext idea had entered CERN, and a young creative engineer had discovered CERN’s pioneering tradition.

It was also by chance that the band of engineers, including Tim, was exposed to novel technology invented by the SPS Controls Group and exported to the PS Controls Group in the late 1970s: remote execution. A program was able to send a small part of itself to another computer and request its remote execution. An extension of this facility came naturally: the possibility for a programmer to include a special call to trigger a remote execution. The PS Control System was one of the first in the world to use Remote Procedure Calls (RPCs) in an operational environment. The RPC expertise created at CERN in the late 70s proved to be essential to the development of Web technology ten years later. Thus, Tim Berners-Lee left CERN in December 1980 with two complementary assets: practical application of hypertext expertise, and exposure to CERN RPC expertise.

### ***CERN’s unique expertise in Document Format***

Tim returned to the UK, where he furthered his knowledge of RPC and more generally of networking. He applied for a fellowship at CERN in 1984 and joined the On-Line Computing Group. Involved in the LEP Data Acquisition System, he proposed using the RPC technology on which he had become an expert. But in parallel, and without any formal mandate, he started to reflect on how to solve the problem of the huge mass of diverse digital information CERN and the HEP

community at large had to deal with. Naturally, he thought of a hypertext system, but unlike Enquire, not a centralized system that, he had realized, would never scale, but a fully distributed system. And the technique to support the decentralized approach was RPC. Hypertext plus RPC: he started to merge the two base technologies he had been working on over the past eight years.

To implement his idea of a document access system, Tim Berners-Lee also needed a standard format for describing text documents, which would be able to efficiently link to other documents. The chance again was that at CERN, world-class expertise existed on document standards. In particular, CERN experts had been major contributors to the Standard Generalized Markup Language (SGML) released by the International Standards Organization (ISO). The Web inventor benefited from the collaboration of the CERN SGML expert to design the Web document format standard, the famous Hypertext Markup Language (HTML).

### ***CERN's unique expertise in Internet Technology***

With HTML, one component of the Web technology was designed. But two others were missing. First a set of communications rules was needed between two computers to exchange documents: how to request a document, how to transport a document if available. Such a set of rules is called a protocol. Tim had to invent a Hypertext Transport Protocol — what he called HTTP. Second, a mechanism was needed to unambiguously and uniquely identify any document or resource; but how to form such a universal and unique identifier, when society is not able to uniquely name human beings on planet Earth? Tim had to invent a Universal Resource Identifier (URI) including a particular type of it to locate the resource, the Universal Resource Locator (URL).

The next chance was that the Web met the Internet. By the mid-1980s CERN started to evaluate and use the emerging TCP/IP protocol, and had gained expertise in Internet technology. By 1988, it became clear that IP (the Internet Protocol) was the protocol of choice for future packet networks. This is just what Tim was looking for. Before IP, the other wide-area technologies (such as IBM's SNA, X.25 or ATM, the Asynchronous Transfer Mode) were based on the connection-oriented principle: before a first packet could be sent, a telephone-like call had to be engaged between any two distant computers. This is not practical for an RPC-based system like the Web that needs very fast interaction. In contrast, IP allows a packet, which may carry a request for a new Web page, to be sent without prior call. Tim Berners-Lee was then able to build his HTTP protocol on the light and efficient Internet Protocol. The chance was also that the Internet is provided with mechanisms unique in the history of telecommunication, one of these being that any system or object connected to the Internet can have a unique name, independent of its real IP address. And in addition, an automatic system (Domain



Name System, DNS) allows the transformation of these names into addresses. To identify documents or resources contained in IP systems, Tim Berners-Lee modelled his URIs/URLs on Internet names.

### ***Keeping it free and protected with open source***

After his first proposal in 1989 (to which he never received a formal reply), Tim started the development that led to the opening of the first server in December 1990. It was again chance that at that time CERN was at the heart of a big star-shaped hub, having the greatest capacity for Internet transfer in Europe. The response time of the CERN Web server was impressive, and collaboration with other servers was boosted by excellent connectivity. Growth was astonishing: the number of Web servers grew from 26 at the end of 1992 to 200 in October 1993.

Then chance intervened yet again. In 1993, Tim Berners-Lee considered it necessary to make Web software more widely available. On 30 April 1993, CERN relinquished all intellectual property rights to the software and put it in the Public Domain. At that time, concepts like open software and public domain were in their infancy. In summer 1994, Tim Berners-Lee left CERN to join MIT and create the World Wide Web Consortium (W3C), keeping a European branch, initially at CERN and then at INRIA. After Tim's departure, the CERN technical team was taken over by new management [55]. The team was preparing the release of version 3 of the CERN server software, but by that time, the principles of open source licencing had become better known at CERN, and the risk of appropriation by third parties was more clearly understood. Indeed, any company could have taken the Web software and (after a minor change) prevented others from using it freely. As luck had it, over those 18 months, this did not happen. In November 1994, version 3 was released under a formal open source licence: CERN retained the intellectual property but gave to anyone the right to use, modify and distribute it freely. To protect freedom and avoid appropriation, all subsequent releases have been distributed by MIT under a similar open source licence.<sup>ε</sup>

### ***Satisfying an unforeseen need***

As stated above, the coming of the Web was a combination of chance and necessity. The Web fulfilled a genuine need — that of the community of particle physicists, made up of more than ten thousand scientists, spread throughout the whole world. The Web was a necessity: the HEP community needed it, but did not know it needed it [55]. This is the wont of a disruptive invention: no-one requests it, but once there, no-one can do without it. Indeed, Tim Berners-Lee's 1989 proposal triggered little interest. One has to admit that re-reading the text, it is hard

---

<sup>ε</sup>An open-source license is a type of license for computer software that allows the source code, blueprint or design to be used, modified and/or shared under defined terms and conditions.

to figure out that it would revolutionize the world. Until 1994, the project received only limited support from CERN [53] but it was allowed to continue, despite the lack of interest from potential users. In 1992 Tim presented the project to the HEPnet Requirements Committee, a gathering of CERN users tasked to formulate the needs in networking. No-one really understood the benefits. Only when they started to use it did the physicists understand that it was an unconscious necessity!

What the Web brought to users is the abolition of login tyranny. Before the Web, it was already possible to access remote information (files, programmes, text, etc.) but on two conditions: first to know and remember on which remote computer the information was stored, and second to be registered on the home computer and to login (or otherwise provide your credentials) on the target computer (with the exception of the special “Anonymous File Transfer” function). The conjunction of clickable hyperlinks and login-less servers removed these two barriers.

### ***Mastering more than one domain***

Beyond chance and necessity, there are other reasons for the success of the World Wide Web. One is the “think-global” spirit. In Tim Berners-Lee’s design, the planetary dimension was present right from the beginning (Fig. 9.9). Even though today’s universality was beyond the wildest dreams, he designed the Web for a global community of physicists, with a permanent concern as to its simplicity and scalability. Also to his merit, Tim Berners-Lee mastered several cutting-edge technical skills, in particular a deep understanding of both networking and document structure techniques. Progress in science and technology is often made by those who excel in more than one domain: in our increasingly complex world, compartmentalization of individuals into specialities can be a brake on creativity. The dual invention by the same person of the Hypertext Markup Language (HTML) and of the Hypertext Transport Protocol (HTTP), two designs that are a perfect match, is a fitting illustration that will leave its mark in history.

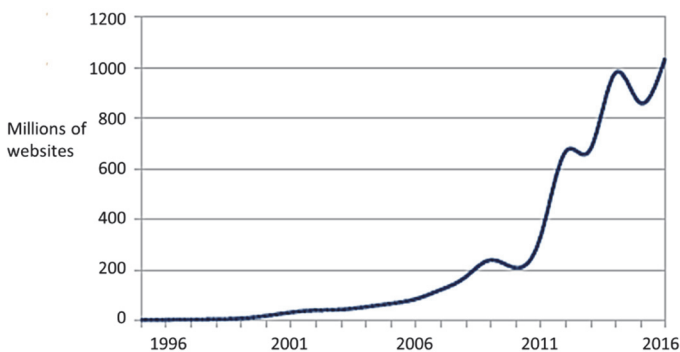


Fig. 9.9. The relentless rise of the Web: number of websites worldwide [56].

## References

1. R. Brun, F. Carminati, G. Galli Carminati, *From the Web to the Grid and Beyond – High Energy Physics Driven Computing Paradigms* (Springer, 2012).
2. CERN Courier, 7, Issue 9, p. 173, September 1967; <https://cds.cern.ch/record/1728889/>.
3. P. Zanella, 30 years of computing at CERN, *12th CERN School of Computing* (1989) pp.1-29; <http://cds.cern.ch/record/205360?ln=en>.
4. H. Blieden *et al.*, in *Informal Meeting on Film-less Spark-chamber Techniques and Associated Computer Use, CERN, Geneva, Switzerland, March 1964*, CERN-1964-030 (CERN, Geneva, 1964) pp. 49-56. <http://dx.doi.org/10.5170/CERN-1964-030.49>.
5. F. Marciano *et al.*, pp. 245-249 in *Informal Meeting on Film-less Spark-chamber Techniques and Associated Computer Use, CERN, Geneva, Switzerland, March 1964*, CERN-1964-030 (CERN, Geneva, 1964). <http://dx.doi.org/10.5170/CERN-1964-030.245>.
6. P. F. Kunz, *Nucl. Instr. & Meth.* **135**, 435 (1976).
7. C. Bertuzzi *et al.*, pp. 329-335 in *Topical Conference on the Application of Microprocessors to High-Energy Physics Experiments, CERN, Geneva, Switzerland, May 1981*, CERN-1981-007. <http://dx.doi.org/10.5170/CERN-1981-007.329>.
8. S. Cittolin and the UA1 Collaboration, *Nucl. Instr. & Meth. A* **263**, 174-179 (1988).
9. D. Castelvecchi, Artificial intelligence called in to tackle LHC data deluge, *Nature* **528**, 18-19 (2015); <http://www.nature.com/news/artificial-intelligence-called-in-to-tackle-lhc-data-deluge-1.18922>.
10. J. Shiers, Data preservation is a journey, *CERN Courier*, May 2016, and references therein, <http://cerncourier.com/cws/article/cern/65032>.
11. T. Bloch, The CERN Computer Centre, *report CERN/DD/74/35* (1974); <http://cds.cern.ch/record/1050339/files/dd-74-35.pdf>.
12. O. Martin and A. Tengvald, Performance measurements on the CDC 7600 at CERN, *report CERN/DD/75/10* (1975); <https://cds.cern.ch/record/307944/files/CM-P00059674.pdf>.
13. G. Lo Presti *et al.*, CASTOR: A Distributed Storage Resource Facility for High Performance Data Processing at CERN, *Proc. 24th IEEE Conf. on Mass Storage Systems and Technologies* IEEE Computer Society (2007) pp. 275-280.
14. B. Segal *et al.*, SHIFT-BETEL: A (very) distributed mainframe, *Proc. Conf. Computing in High Energy Physics '94*, San Francisco, California (1994) pp. 188-191.
15. J.-P. Baud *et al.*, Mainframe services from Gigabit- networked workstations, *Proc. Summer 1992 USENIX Conference*, June 1992, San Antonio, Texas, U.S.A.; <http://usenix.org/legacy/publications/library/proceedings/sa92/ baud>.
16. A. Fucci *et al.*, pp. 341-354 in *Topical Conf. on the Application of Microprocessors to High-energy Physics Experiments, CERN, Geneva, Switzerland, May 1981*, CERN-1981-007 (CERN, Geneva 1981). <http://dx.doi.org/10.5170/CERN-1981-007.341>.
17. J.-P. Baud *et al.*, SHIFT, the scalable heterogeneous integrated facility for HEP computing, *Proc. International Conference on Computing in High Energy Physics, Tsukuba, Japan*, Universal Academy Press, Tokyo (1991) pp. 571-582.
18. M. Aderholz *et al.*, Models of networked analysis at regional centres for LHC experiments (MONARC), *report CERN/LCB 2000-001*(2000), <http://cds.cern.ch/record/510694/files/Phase2Report.pdf>.
19. I. Foster and C. Kesselman. *The GRID: Blueprint for a New Computing Infrastructure*, (Morgan Kaufmann Publishers, San Francisco, 1998).

20. F. Gagliardi *et al.*, The EU DataGrid project: building and operating a large scale grid infrastructure, in *Engineering the Grid: Status and Perspective*, eds. B. Di Martino, J. Dongarra, A. Hoisie, L.Y. Yang, and H. Zima (American Scientific Publishers, 2006); <http://www.gridpp.ac.uk/papers/Perf2002Paper.pdf>.
21. L. Robertson, *Computing Services for LHC – from Clusters to Grids*, Chapter 3 of ref. [1].
22. H. Slettenhaar, Data communications at the CERN Computer Centre, *report DD/74/26* (1974); <http://cds.cern.ch/record/418287/files/dd-74-26.pdf>.
23. J.M. Gerard *et al.*, The use of the FOCUS system to monitor an experiment at the CERN Intersecting Storage Rings, *report DD/73/13* (1973); <http://cds.cern.ch/record/1050741/files/dd-73-13.pdf>.
24. C. Parkman and G. Lee (eds.), *OMNET — High Speed Data Communications for PDP-11 Computers*, CERN-1979-011 (CERN, Geneva, 1979). <http://dx.doi.org/10.5170/CERN-1979-011>.
25. T. Bruins *et al.* (eds.), *SUPERMUX — A Multi-Host Front End Concentrator System for Asynchronous Consoles*, CERN-1974-019 (CERN, Geneva, 1974). <http://dx.doi.org/10.5170/CERN-1974-019>.
26. J.M. Gerard (ed.) *CERNET — a High-Speed Packet-Switching Network*, CERN-1981-012 (CERN, Geneva, 1981). <http://dx.doi.org/10.5170/CERN-1981-012>.
27. B.M. Segal, A Short History of Internet Protocols at CERN (1995); <http://cnlart.web.cern.ch/cnlart/2001/001/tcpip/>
28. P. G. Innocenti, The LEP control system: architecture, features and performance, *Proc. 14th Int. Conf. on High-Energy Accelerators*, Tsukuba, Japan, (1989) pp. 885-892; <http://cds.cern.ch/record/210879/files/p183.pdf>.
29. R. J. Laukner, R. Rausch, SPS and LEP controls: status and evolution towards the LHC era, *Proc. Int. Conf. on Accelerator and Large Experimental Physics Control Systems*, Chicago, U.S.A. (1995); <http://ccdb5fs.kek.jp/cgi-bin/img/allpdf?199603225>.
30. J. Serrano *et al.*, The white rabbit project, *Proc. INIB2013*, Oxford, UK (2013); <http://accelconf.web.cern.ch/AccelConf/IBIC2013/papers/thbl2.pdf>.
31. R. Beach, UltraNet: An Architecture for Gigabit Networking, *Proc. 15th Conf. on Local Area Networks*, IEEE (1990) pp. 232–248.
32. Ben Segal *et al.*, HiPPI in the CERN Computer Centre, CERN Report CN/96/13; <http://cds.cern.ch/record/311057/files/cn-96-013.pdf?version=1>.
33. CERN Press Release, <http://press.cern/press-releases/2000/09/cern-first-demonstrate-10-gigabitsec-network-four-major-computer>.
34. P. Grun, Introduction to Infiniband for end users, *Infiniband Trade Association*, April 2010; <https://cw.infinibandta.org/document/dl/7268>.
35. O. Martin, *The hidden Prehistory of European Research Networking*, E-Book (Trafford Publishing, 2012); <http://oliviermartin.authorsxpress.com/author/oliviermartin/>.
36. F. Fluckiger, CERN celebrates another key contribution to the Internet, *CERN Computer Newsletter*, June issue (2009); <http://cerncourier.com/cws/article/cnl/39610>.
37. H. Davies and B. Bressan, *A History of International Research Networking* (Wiley, 2010); <http://eu.wiley.com/WileyCDA/WileyTitle/productCd-352732710X,subjectCd-CM12.html>.
38. F. Fluckiger, The European researchers' network *La Recherche*, **328** (2000) (*translation*); [https://cern.ch/Fluckiger/Articles/F.Fluckiger-The\\_European\\_Researchers\\_Network.pdf](https://cern.ch/Fluckiger/Articles/F.Fluckiger-The_European_Researchers_Network.pdf).
39. <https://root.cern.ch/tmva>.

40. T. Sjöstrand *et al.*, An introduction to PYTHIA 8.2, *Comput. Phys. Commun.* **191**, 159 (2015); <http://home.thep.lu.se/~torbjorn/pdfdoc/pythia8200.pdf>.
41. R.L. Ford and W. R. Nelson, The EGS Code System, *report SLAC-210*, UC-32 (1978); <http://slac.stanford.edu/pubs/slacreports/reports03/slac-r-210.pdf>.
42. R. Brun *et al.*, Geant3 user's guide (1987); <http://cds.cern.ch/record/1119728?ln=en>.
43. S. Agostinelli *et al.*, Geant4, a simulation toolkit, *Nucl. Instr. & Meth.* **506** (2003).  
J. Allison *et al.*, Geant4 developments and applications, *Trans. Nucl. Science* **53** (2006).
44. A. Ferrari *et al.* (eds.), *FLUKA: a Multi-Particle Transport Code*, CERN 2005-010 (CERN, Geneva, 2005); <http://dx.doi.org/10.5170/CERN-2005-010>.
45. R.K. Bock and J. Zoll, Bubble chamber experiments: central computers in the analysis, *CERN Courier* **12**, issue 3 (1972); <http://cds.cern.ch/record/1729464/files/vol12-issue3.pdf>.
46. R. Brun *et al.*, PAW - A general purpose portable software tool for data analysis and presentation, *Computer Physics Communications* **57**, 432-437 (1989).
47. I. Antcheva *et al.*, ROOT - A C++ framework for petabyte data storage, statistical analysis and visualization, *Computer Physics Communications* **180**, 2499-2512 (2009).
48. CERN Program Library, <http://cernlib.web.cern.ch/cernlib>.
49. R. Brun and D. Lienart, HBOOK user guide, version 4, *CERN program library Y250* (1988).
50. F. James, MINUIT. Function minimization and error analysis, Reference manual, *CERN Program Library, long write-up D506* (1994); <https://root.cern.ch/d35c7d8c.web.cern.files/minuit.pdf>.
51. R. Brun *et al.*, ZBOOK – User guide and reference manual, *CERN program library Q210* (1984).
52. J. Zoll, PATCHY reference manual: version 5.04 (1995); <http://cds.cern.ch/record/446554?ln=en>.
53. B.E. Carpenter, *Network Geeks, How they Built the Internet* (Springer, 2013); <http://www.springer.com/us/book/9781447150244>.
54. J. Gillies, R. Cailliau, *How the Web Was Born, The Story of the World Wide Web* (Oxford University Press, 2000).
55. D. Raggett, My involvement with the early days of the Web, *W3C 10th Anniversary Conference*, Boston (2004); <http://www.w3.org/People/Raggett/the-early-days-of-the-Web.htm>.
56. <http://news.netcraft.com/archives/2016/05/26/may-2016-web-server-survey.html>.