CM-P00060644

CERN-ECT 92-8 MC

J09524

ATION FOR NUCLEAR RESEARCH

CERN/ECP 92-8 Rev.

31 August 1992

(Revised on 21 December 1992)

THE DATA ACQUISITION SYSTEM OF THE OPAL DETECTOR AT LEP

J.T.M. Baines¹, F. Beck², H.J. Burckhart², D.G. Charlton², R. Cranfield³, G. Crone³,
P.A. Elcombe⁴, P. Farthouat², C. Fukunaga^{5,a}, N.I. Geddes⁶, C.N.P. Gee⁶, F.X. Gentit⁷,
W. Gorn⁸, J. Hart^{4,b}, J.C. Hill⁴, S.J. Hillier⁹, B. Holl^{2,c}, R.E. Hughes-Jones¹, R. Humbert¹⁰,
M. Jimack¹¹, R.W.L. Jones¹², C. Kleinwort^{2,d}, F. Lamarche¹³, P. Le Du⁷, D. Lellouch¹⁴,
L. Levinson¹⁴, A.J. Martin¹², J.P. Martin¹³, F. Meijers², R.P. Middleton⁶, A. Montanari¹¹,
B. O'Neill⁸, G. Quast¹⁵, J. Raab^{2,e}, D. Robinson², O. Schaile¹⁰, P. Scharff-Hansen², P. Schenk¹⁶,
H. von der Schmitt¹⁷, P. Sherwood³, R. Shypit¹⁸, G.P. Siroli¹¹, T.J. Smith², R. Sobie^{16,f},
R.W. Springer¹⁹, D. Strom^{20,g}, T. Tsukamoto⁵, A.T. Watson⁹, N.K. Watson², P.S. Wells²,
S. Wotton^{17,h} and W. Zeuner²

Abstract

This report describes the 1991 implementation of the data acquisition system of the OPAL detector at LEP including the additional services and infrastructure necessary for its correct and reliable operation. The various tasks in this "on-line" environment are distributed amongst many VME subsystems, workstations and minicomputers which communicate over general-purpose local area networks and special-purpose buses. The tasks include data acquisition, control, monitoring, calibration and event reconstruction. The modularity of both hardware and software facilitates the upgrading of the system to meet new requirements.

Submitted to Nuclear Instruments and Methods in Physics Research

- ¹ University of Manchester, Schuster Laboratory, Dept of Physics, UK.
- ² CERN, European Organization for Nuclear Research, Geneva, Switzerland.
- ³ University College London, London, UK.
- ⁴ Cavendish Laboratory, Cambridge, UK.
- ⁵ University of Tokyo, Intern. Centre for Element. Part. Phys. and Dept of Physics, Tokyo; and Kobe University, Kobe Japan.
- ⁶ Rutherford Appleton Laboratory, Chilton, Didcot, Oxfordshire, UK.
- ⁷ CEN Saclay, DPhPE, Gif-sur-Yvette, France.
- ⁸ University of California, Department of Physics, Riverside, California, USA.
- ⁹ University of Birmingham, School of Physics and Space Research, Birmingham, UK.
- ¹⁰ Albert Ludwigs Universität, Fakultät für Physik, Freiburg, Germany.
- ¹¹ Università di Bologna and INFN, Dipartimento di Fisica, Bologna, Italy.
- ¹² University of London, Queen Mary and Westfield College, London, UK.
- ¹³ Université de Montréal, Lab. de Phys. Nucl., Montréal, Québec, Canada.
- ¹⁴ Weizmann Institute of Science, Nucl. Phys. Dept, Rehovot, Israel.
- ¹⁵ Universität Hamburg/DESY, II. Inst. für Experim. Physik, Hamburg, Germany.
- ¹⁶ University of Victoria, Dept of Physics, Victoria BC, Canada.
- ¹⁷ Universität Heidelberg, Physikalisches Institut, Heidelberg, Germany.
- ¹⁸ University of British Columbia, Dept of Physics, Vancouver Canada.
- ¹⁹ University of Maryland, Dept of Phys. and Astronomy, College Park, Maryland, USA.
- ²⁰ University of Chicago, Enrico Fermi Inst. and Dept of Physics, Chicago, USA.

^a Now at Meiji Gakuin University, Yokohama, Japan.

^b Now at CERN, European Organization for Nuclear Research, Geneva, Switzerland.

^c Now at Kraftwerks-Simulator GmbH, Essen, Germany.

^d Now at DESY, Hamburg, Germany.

^e Now at Johannes-Gutenberg Universität Mainz, Inst. für Physik, Germany.

^f And IPP, McGill University, High-Energy Phys. Dept, Montréal, Québec, Canada.

^g Now at University of Oregon, Dept of Physics, Eugene, Oregon USA.

^h Now at Universität Heidelberg, Physikal. Inst., Heidelberg, Germany.

1 INTRODUCTION

This report describes the data acquisition (DAQ) system of the OPAL detector [1, 2] at LEP. The main elements of the OPAL detector are a central tracking system in a solenoidal coil, a time-of-flight detector, an electromagnetic calorimeter system, a hadronic calorimeter system, a muon detection system and forward detectors for luminosity measurement (fig. 1). These components are subdivided further into the subdetector elements listed in table 1. The subdetectors were designed and tested independently, to a large extent at institutes away from CERN, and were then assembled to form the complete detector at CERN.

LEP is a synchrocyclotron designed to accumulate and accelerate four bunches of electrons and four counter-rotating bunches of positrons. In the present LEP configuration, the bunches are stored at centre-of-mass energies corresponding to the peak of the Z^0 resonance (~ 91 GeV). The electron and positron bunches are steered and synchronized so that they intersect only at a few places on the LEP circumference and with a bunch-crossing interval of $22 \mu\text{s}$. The OPAL detector is situated at one such place in order to detect the products of any electron-positron collisions. LEP will continue to operate throughout most of the 1990s. During this time it will be upgraded to provide higher luminosity and higher maximum beam energy. The number of electron and positron bunches will be increased resulting in a reduced time between bunch crossings.

At the peak of the Z^0 resonance and at the LEP design luminosity of $1.6 \times 10^{31} \text{ cm}^{-2} \text{ s}^{-1}$, Z^0 decays are detectable at a rate of ~ 0.55 Hz with a ratio of hadronic to non-hadronic decays of $\sim 7:1$. Bhabha scattering within the acceptance of the small-angle luminosity detectors contributes a further 1.5 Hz to the interaction rate. Readout of the complete detector is triggered by a system having a high efficiency for selecting bunch crossings which give rise to the above interactions [3]. The rate of spurious triggers is such that the first-level trigger rate is expected to be < 10 Hz at the LEP design luminosity. The amount of data generated per trigger amounts to several megabytes after digitization but data processing, before recording, reduces it to an average of 46 kbytes for hadronic Z^0 decays and typically 12 kbytes for other triggered "events".

Data acquisition for the detector requires a bandwidth and data reduction capability which cannot be provided by a single central computer system. Furthermore, in order to benefit from any future improvements in LEP performance and to accommodate upgrades of the OPAL detector, the DAQ system must be easily adaptable. A modular, distributed DAQ system based on commercially available microprocessor systems, workstations and minicomputers was therefore chosen.

The complete on-line system has been operational since LEP was commissioned in 1989. The following sections give a comprehensive description of all its major components as it was used in 1991. Section 2 gives an overview of the total system. Section 3 gives a general description of the VME-based subdetector DAQ systems. Section 4 describes the components of the central on-line system involved with the data flow from the subdetectors to the recording of raw and fully reconstructed data. Section 5 describes other essential subsystems used, for example, for monitoring, calibration and DAQ control. Section 6 summarizes the overall performance and outlines the upgrade plans. Additional material on the related infrastructure is given in Appendices A, B and C.

2 OVERVIEW

The primary function of the on-line system is data acquisition which involves selecting, collecting, processing and storing the raw data from the subdetectors' digitizers. The complete on-line system also includes:

- DAQ control, to coordinate all activities in common data taking and to give a single operator full control over the entire DAQ system.
- Monitoring, to verify the performance of the detector and the quality of the data during data acquisition.
- Event reconstruction, to convert the raw event data into quantities used in subsequent physics analyses.
- Management of the calibration constants, to be used by the event reconstruction programs.
- "Slow control", to supervise the operational parameters of the detector and to monitor safety-related quantities.
- Communication with the LEP control computers, to provide automatic exchange of monitoring information between LEP and OPAL.

2.1 Design criteria

No standardization of the readout of the subdetector electronics is imposed because subdetector groups have distinct requirements and expertise. However, all subdetectors use a dedicated computer for their data acquisition which is required to have standard hardware and software interfaces with the central system.

The subdetector computers must be able to operate autonomously for development, commissioning and calibration and to operate together for physics data taking. This dual requirement influences much of the design of the data acquisition components.

All subdetectors must perform the digitization and readout as quickly as possible since no further events can be selected during this "dead time". The complete system must have sufficiently high throughput and have high reliability in order to achieve a high overall data taking efficiency.

The integrated DAQ system must be controllable by a single operator who may have little detailed knowledge of the system. To achieve this, the individual components must be highly reliable and data flow monitoring and recovery procedures must enable the operator to diagnose and recover quickly from the occasional hardware or software malfunctions. In case of failure, it must be possible to reset a subsystem and bring it back into operation without disturbing other subsystems. Remote access to the monitored information is required because of the size of LEP; the OPAL detector is located 100 m underground and ~ 10 km from the office buildings at the CERN main site.

With LEP operating almost continuously over several years, the issues of data flow and storage are a major concern. Event reconstruction must keep pace with data taking and should, if possible, be part of the on-line system. The raw data must be stored in a compact format to minimize the data handling requirements both for storage and for later reprocessing.

Calibration data must be available during data taking to allow prompt event reconstruction. However, more refined calibrations become available only after a first analysis. Therefore, it must be possible to update the calibration database from both the on-line and the off-line computers.

The complexity and size of the detector necessitate a dedicated, computerized, slow-control system in order to ensure the safe operation of the detector. It must continuously monitor all operational parameters with the possibility of automatic corrective actions and it must provide the operator with a coherent interface to the whole detector.

2.2 Implementation

The on-line system is shown schematically in fig. 2. The lowest layer consists of the subdetector and trigger computers operating in parallel. These are all implemented in VME (Appendix A.1) and are called the local system crates (LSCs). The trigger system selects events based on fast signals from some of the subdetectors. If an event is selected, it broadcasts a readout request over a dedicated trigger bus to all LSCs. The “event builder” then synchronizes and merges the event data which it collects from the subdetectors. The event builder forwards the events to the “filter” [4] which refines the trigger selection, compresses the associated data, and performs extensive event monitoring. Events selected by the filter are saved in a large disk buffer on the filter system. This buffer has a capacity corresponding to several hours of data taking and decouples the data acquisition from event reconstruction and data recording. The event reconstruction system fetches events from the filter buffer over the Apollo Token Ring [5], records them on optical disks for long-term storage, and reconstructs the events as soon as up-to-date calibration data are available. The reconstructed events are then sent over Ethernet to other systems used for physics analysis. All the activities in the LSCs, the event builder and the filter are controlled by a central DAQ control system.

A heterogeneous selection of computers is used in this implementation, the best solution having been chosen for each component at the time the component was implemented or upgraded. All components with real-time requirements are implemented in VME (Appendix A.1). Those with computationally or memory-intensive requirements such as filter and event reconstruction are implemented on HP Apollo DN10000 RISC-based workstations [5]. DAQ control and several other services are implemented on a VAXcluster [6]. Not shown in fig. 2 are some Macintosh-II [7] computers, which are used for interactive graphics applications such as event display, histogram presentation and operator interface to the slow control system.

The VMEbus family (VMEbus, VSB and VICbus (Appendix A.1)) was chosen as the unifying standard for the subdetectors’ dedicated DAQ buses and the central DAQ system. The VMEbus has interfaces to commonly used DAQ buses (CAMAC [8] and Fastbus [9]) and the VICbus provides efficient interconnection between VME crates as well as interfaces to other buses commonly used in workstations and personal computers. The LSCs have a VICbus connection to the event builder which has a VICbus connection to the filter. Each LSC comprises: at least one single-board computer with an MC680x0 CPU [10], a VICbus interface and a trigger bus interface. Other modules may be used to meet specific needs. The subdetector LSCs vary from single-processor systems to multicrate multiprocessor systems according to the complexity of the readout and data processing but this is hidden from the central DAQ system through standard hardware and software interfaces.

The OS-9 (Appendix B) is used as the real-time operating system for all single-board computers

in VME. OS-9 provides a single-processor, real-time, multitasking, multiuser environment with a UNIX-like [11] shell and file system. It provides program development facilities and has sufficient capabilities to allow the subdetector computers to act autonomously. Multiprocessor applications in OS-9 can use named pipes and shared data modules for interprocess communication.

The Ethernet local area network (LAN) [12] connects all computers and all VME crates. Standard high-level protocols (Appendix C.1) are used for communication between different operating systems. Each LSC subdetector therefore has three hardware connections to the central system: trigger bus, VICbus, and Ethernet. Almost all communications between the central system and the LSCs take place over Ethernet. There are two exceptions for reasons of speed: trigger information is broadcast via the trigger bus; and event data are transmitted via VICbus.

3 SUBDETECTOR DATA ACQUISITION

Each subdetector DAQ system can operate both as part of common data taking and in autonomous mode. It combines the following functions:

- Handling of the trigger interrupt, readout of the digitizers and subsequent data reduction, formatting and partial event reconstruction.
- Control functions, e.g. for high voltage.
- Monitoring the correct operation of the subdetector.
- Automatic and operator-driven calibration procedures: either parasitically with common data taking or in dedicated runs.
- Autonomous data taking with local recording.

Each subdetector is equipped with at least one LSC with interfaces to both subdetector-specific digitizing electronics and to the central DAQ system. Because of the layout of OPAL, a number of subdetector DAQ systems are split, having an LSC on opposite sides of the detector.

3.1 Digitizing electronics

The subdetectors have widely differing detector hardware. The kind of digitizing electronics used, number of channels and resulting raw digitized event size are given in table 2.

For the digitization of drift chamber signals, flash ADCs (FADCs) are often used. The jet chamber, z-chambers and forward drift chambers use DL300 FADCs [13], which are interfaced to VSB. For the muon barrel detector drift chambers, Fastbus-based FADCs [14] are employed. The vertex detector drift chamber is read out by a purpose-built TDC system [15].

The calorimetric detectors use charge integrating ADCs (CIAs) to digitize the analog signals. There is a Fastbus [16, 17] and a CAMAC-based [18] implementation. Each channel is digitized by a 12-bit ADC with two sensitivities which are different by a factor of eight, ensuring a 15-bit dynamic range. The CIA modules are also equipped with fast, analog sum circuits whose output summed signals are used for triggering purposes.

A number of subdetectors employ an analog multiplexing scheme in order to read out a large number of channels economically and to reduce the required number of cables. Sample-and-hold units, which are directly mounted onto the detector, integrate and store the charge of each detector channel in groups of 32 [19]. A VME sequencer module controls the multiplexed transfer of these charges into a VME 12-bit ADC module. The digitization may also be done at two sensitivities and the results stored in a local RAM. In this way, one ADC module digitizes the data of many channels and serves also as a single-event buffer.

The silicon microvertex subdetector performs an analog multiplexed readout with Fastbus SIROCCO IV modules [20]. The sampling of the signals from the silicon strips is done by VLSI chips [21] bonded to the detector. The SIROCCO IV module performs the digitization and stores the data in one of four buffers accessible by a digital signal processor (DSP).

For subdetectors which comprise only a few hundred channels, standard CAMAC ADCs or TDCs are used. Subdetectors whose channels do not contain analog information use dedicated electronics based on shift registers which produce either a bitmap of active detector elements or a list of addresses of such elements.

3.2 The local system crate

Some LSC components are common to all subdetectors but front-end interfaces and the number of single-board computers in the LSC are adapted to individual needs. The subdetector-specific LSC software is built on common libraries and utilities.

3.2.1 Hardware

A generic LSC is shown in fig. 3. It contains at least one single-board computer [22, 23] with optional hard disk and the VME modules which interface to Ethernet, VICbus and trigger bus. The local trigger unit (LTU) is the interface to the trigger bus. It can operate in common mode, receiving triggers from the trigger bus, or in autonomous mode using locally-generated triggers. The readout electronics, which reside in a variety of bus standards including VME, Fastbus, CAMAC and private designs (table 2), are interfaced via VME or, more efficiently, via the single-board computer's VSB. The most often used interfaces are described in Appendix A.2. The single-board computer's memory is usually supplemented with an additional VME RAM module [24].

The event builder's VICbus connects to one LSC per subdetector. Split subdetectors with an LSC on each side of the detector use a dedicated VICbus connection between the two crates. This is used by the LSC connected to the event builder to merge the subevent data from the two sides before transmission to the event builder. The trigger bus connects to both LSCs to allow the two sides to be read out in parallel.

Additional LSC modules may include interfaces to auxiliary systems such as high-voltage or low-voltage control systems or to storage devices such as Exabyte [25] 8 mm tape drives for local data recording. Many subdetectors also connect their LSC to a MicroVAX [6] or Macintosh-II computer for monitoring or histogram presentation.

3.2.2 Software

The LSC software comprises the OS-9 operating system and the subdetector-specific readout, data processing, monitoring and control software. Each LSC runs a number of processes for network communication with central services (sect. 5). Programming languages include FORTRAN (Appendix B.4), C and occasionally 68 k assembler.

OS-9 device drivers were developed for special data acquisition devices, in particular for the LTU whose driver responds to VME interrupts derived from signals on the trigger bus. On receiving a trigger, a “busy” signal is asserted which must be cleared by software after readout has completed. When several processors participate in the readout, the trigger signal is redistributed within the crate.

The readout task is not handled within the LTU driver although this would be most efficient. The driver instead increments an OS-9 event which causes OS-9 to schedule the readout task, usually with priority above time slicing. This allows the readout task to be written as a normal user program, which can perform all I/O and is much easier to write and test. The readout must complete as fast as possible in order to be ready for the next trigger. Readout, data compression, formatting and transmission to the event builder are therefore normally handled by separate tasks, which all use a buffer manager for temporary storage of event data in memory. Some LSCs need several stages of event buffering. A buffer manager which allows multiple stages, one producer and multiple consumers per stage, was developed for this purpose. The output buffer is normally configured to reside in the triple-ported memory of the VICbus interface. The LSC produces subevents in the ZEBRA format required by the event reconstruction program (Appendix C.2).

Much of the common DAQ software is maintained as a suite of programs called the DAQ “skeleton” [26]. This collection of service programs, libraries and utilities provides a standard framework adaptable to the needs of each LSC. The skeleton includes the buffer manager and all the OS-9 tasks that handle event transmission between subsystems. It also covers communication with the central DAQ control as well as subroutine interfaces to the local trigger unit and the error message system. This common framework can be centrally maintained and helps development and trouble shooting through shared knowledge about important aspects of the DAQ system.

3.3 Readout and data processing

A considerable reduction in data volume is achieved by pedestal subtraction and zero suppression. For the drift chamber DL300 FADC readout, the zero suppression is done in hardware. This reduces the data volume typically by more than a factor of ten and allows the readout by VME processors to be completed within a short dead time. Similarly, the Fastbus FADC modules used in the readout of the muon barrel drift chambers contain hardware hit finders to identify the words above pedestal. The other subdetectors perform the pedestal subtraction and zero suppression in software, often by the LSC processor. This is done either asynchronously by a data compression task after all data are read out and buffered or in some cases by the readout task itself. In the case of the silicon microvertex subdetector, subtraction of continuously updated pedestal values is performed by the DSPs in the SIROCCO IV modules.

The nominal dead time due to digitization and readout is given for all subdetectors in table 3. The dead time can depend on event size and trigger rate and can be influenced by other activities in

the LSC processor. For a given trigger the overall dead time is determined by the last subdetector to complete readout (sect. 4.1) and ranges from 8 ms to > 100 ms with a mean value of 15 ms.

In addition to the data reduction by pedestal subtraction and zero suppression, further data processing is done by the main LSC computer or by one or more of the general-purpose processors listed in table 3. All drift chamber type subdetectors with FADC readout process the raw data to resolve multiple hits and to derive their space coordinates and charge. In order to reduce the data volume, the raw data are subsequently discarded except when needed for test and monitoring purposes. In the case of the jet chamber and muon barrel the data are analysed further to reconstruct tracks. For the jet chamber, tracks are found in three dimensions using a fast chaining method [27]. The track fit yields three vectors and track charge information. The jet chamber uses 20 MC68040 CPUs for FADC pulse shape analysis and track reconstruction. These tracks are subsequently used by the filter. For the muon barrel the points obtained in the four drift chamber layers are associated to form track segments. Some of the calorimetric and presampler subdetectors perform cluster analysis and conversion to physical units. The muon endcap subdetector readout identifies hits using DSPs which cooperate with three MC68030 processors to further reduce the data.

The complexity of the data reduction done in the LSC is subdetector-dependent. The maximum event throughput depends on the time required to process the events and on the number of single-board computers available in the LSC. The slowest LSC limits the throughput of the whole system. Subdetectors with extensive processing therefore use multiple single-board computers in their LSC. Even so, the subdetectors' maximum throughput rate ranges from 5 to 25 events/s. The time required for a subevent to move from the first readout buffer to the LSC's final output buffer fluctuates with the complexity of individual subevents. Subdetectors which implement processing on multiple processors in parallel may even deliver events to the output buffer out of sequence.

After data processing and formatting and while awaiting collection by the event builder the subevents are buffered at the LSC. The average subevent sizes for hadronic Z^0 decays and other event types are indicated in table 3. It results in an average total size of 211 kbytes for hadronic events and 60 kbytes for other types of events. Occasionally, events exceed 1 Mbyte in size.

4 CENTRAL DAQ COMPONENTS

Common data taking involves the operation of the subdetector DAQ systems together with the trigger system, the event builder, the filter and the event reconstruction system as outlined in sect. 2.2; this section also describes the function of each of the central components.

4.1 The trigger system

The decision whether to read out on a given bunch crossing is taken by the central trigger logic without introducing dead time. This decision is based on promptly available signals provided by the central vertex and jet chambers (via the track trigger subsystem), the time-of-flight detector, the electromagnetic and hadronic calorimeters, the muon detectors and the forward detectors. The central trigger system hardware is installed in a purpose-built crate. A description of the OPAL trigger system including details of the hardware and trigger decision logic can be found in ref. [3].

The global trigger unit supervises the subdetector readout synchronization and timing. It broadcasts the trigger decision to the subdetectors on the trigger bus to which LTUs in each subdetector readout crate are connected. In case of a negative trigger decision, a reset pulse is distributed in time for the subdetectors to be ready for the next bunch crossing. This time is needed by the slowest subdetector to clear its front-end electronics in order to be ready for the next bunch crossing. If an event is accepted, the global trigger unit generates a trigger pulse and broadcasts the trigger number of this event and the trigger pattern (19 bytes in total) to the LTUs. The trigger number is used later as a key when merging subevents in subsequent stages of the data acquisition. On receipt of a trigger, the subdetectors initiate digitization and each LRU generates a VME interrupt and asserts the busy signal on the trigger bus. The readout process in each LSC clears its busy signal at the end of its readout. New triggers may be generated only after all LSCs have cleared the busy signal. Failure of a subdetector to release the busy signal after a time-out period of 4 s leads to an error condition reported by the trigger LSC.

The DAQ control system supplies the relevant run parameters to configure the trigger hardware logic. The subdetector input signals to the trigger logic are included in the event record and can thus be compared with the information read from the subdetectors in the subsequent filter stage or off line.

4.2 The event builder

The event builder collects subevents from the individual sources, merges these, and forwards the full events to one or more destinations. There are 17 sources: one LSC per subdetector, and one destination: the filter. All are connected to the event builder via VICbus. In addition to the event-related data from the subdetectors, a small amount of data from the slow-control system is read out for each event.

Subevents corresponding to the same trigger are not available at the same time in all subdetectors because of the variable processing time in each LSC (sect. 3.3). The event builder must therefore read and buffer subevents in the order they become available in the LSC's output buffers and must allow LSCs to deliver their subevents out of sequence. The event builder must have sufficient buffer memory to allow for latencies of several seconds, which may arise from extensive data processing in the LSCs. Subevents are merged into full events using the trigger number as a key and the merged events are output in trigger number sequence. Dynamic memory allocation is required since events span a large range in size. The limit is set to 2 Mbytes per event.

Out-of-sequence subevents can lead to deadlock situations if the buffer space is exhausted before a missing subevent arrives. This is controlled by a direct feedback from the event builder to the trigger, which allows the event builder to inhibit further triggers when buffer space becomes critical.

The DAQ control system can define the set of LSCs to be included in the event building process. This can be updated at any time the run is paused by the operator or by an error condition. Individual subdetectors can thus be excluded or included at any time, a feature which is used during error recovery.

Extensive error handling is provided in order to achieve a high overall data taking efficiency. For example, an error condition is generated if a subevent does not arrive within a defined time-out. The DAQ control system is notified and corrective action can then be taken, automatically

or through operator intervention. A resynchronization procedure allows quick recovery in case a reset of an LSC, the event builder or the filter is required, avoiding the necessity to reset any of the others.

Transmission of a subevent is controlled by an event descriptor message (containing trigger number, subevent address and length) and an event acknowledge message. The LSC sends an event descriptor message to the event builder. The event builder uses this information to initiate a DMA transfer of the subevent from the LSC. On completion of the transfer it sends an acknowledge message to the LSC. The same mechanism is used for event transfers between event builder and filter. The event descriptor and acknowledge messages are transmitted over VICbus by means of an interrupt-driven VICbus device driver. The DMA transfer from the output buffer of the remote crate uses memory-mapped access over VICbus.

The data flow through the event builder is controlled by a single OS-9 process implemented with the object-oriented language C++ [28]. It communicates with a process in each source and destination system and initiates DMA data transfers. The OS-9 signals are used to support interrupts and alarms. The OS-9 alarms provide a convenient way of implementing time-outs. A C++ object method is invoked either by other objects, or after the occurrence of a particular signal. Hence each occurrence of an external event, e.g. the availability of the next subevent, completion of a DMA data transfer, or a command from the DAQ control system leads to the execution of an object method.

The strict separation of control and event data transfer in the event builder allows the hardware configuration to be adapted to meet the required throughput. The event builder assembly is shown in detail in fig. 4. The LSCs are grouped into four VICbus branches. The cable lengths vary from a few metres to 100 m with transfer rates up to 6 Mbytes s^{-1} on the shortest distances. The functions of control, data copying and data formatting are distributed to a number of single-board computers, and two 16-Mbyte dual-ported VME/VSB memories are used for multi-event buffers. The supervisor processor controls the event building process. The copy processors perform a DMA operation in which the subevent data from the LSCs are read via VSB and written into the event input buffer via VME. Parallelism is provided by a pair of copy processors connected separately by VSB to VICbus branches 1 and 2-4, respectively. This partition is motivated by the fact that about half of the data are generated by the jet chamber, which is the only subdetector on branch 1. After all parts of an event are received, the merge processor copies the event pieces from the input buffer into contiguous memory in the output buffer and formats the assembled event. This processor accesses the multi-event buffers over the VSB port. Finally, a third processor copies events from the output buffer to the triple-ported memory of the filter's VICbus interface. In this configuration the event builder can handle a sustained rate of 25 Hz for 100-kbyte events.

4.3 The filter

The filter [4] acts as a second-level software trigger. Here the full events are available and can be classified into physics categories. Backgrounds that are accepted through the first-level hardware trigger in order to obtain a high and easily measurable efficiency are then further reduced. Under 1991 conditions, $\sim 35\%$ of the events were rejected.

The event classification requires a partial event reconstruction. Use is made of clusters in the electromagnetic calorimeters and the forward detector calorimeter, hits and track segments in the muon detectors and tracks in the jet chamber. Some partial reconstruction is done already at the

LSC level: it is complemented by the filter. The classification of events also provides well-defined samples for the immediate monitoring of detector and trigger hardware. General monitoring of the data quality using histograms is carried out to supplement any such monitoring in the LSCs. In addition, possible trigger logic failures are identified by checking the consistency of the event data with the observed trigger classification for each event. Any anomalies are reported to the DAQ operator. Further monitoring in the form of an on-line event display (sect. 5.4.4) provides a powerful tool to identify problems quickly.

An HP Apollo DN10000 RISC-based workstation is used to perform these functions. It has a powerful programming environment, substantial CPU resources and interfaces to VME. The filter workstation is configured with four CPU boards, 64 Mbytes of main memory and four 700-Mbyte disks and contains interfaces to both Ethernet and Apollo Token Ring networks. It runs the Domain/OS operating system, which provides the Apollo Aegis system, and full implementations of both SysV.3 [29] and BSD4.3 [30] UNIX environments.

The DN10000's peripheral VMEbus hosts a VICbus module for connection with the event builder. A device driver supports interrupt handling and memory-mapped access to the VME/VIC address space. The filter communicates with central services over TCP/IP via Ethernet and cooperates, using event-builder buffer memory, with two Macintosh-II computers running a histogram presenter and the on-line event display.

The event processing is divided between several processes operating in parallel. The MODEL buffer manager [31] is used to coordinate the event handling.

A small amount of data is added to the event record, including filter analysis results and LEP run parameters, accessible via the network data base (Appendix C.3.3). Each event record includes an event header with 64 words of basic event information. Only this event header is kept for rejected events. Data compression is performed on the accepted events, reducing their data volume by a factor five (Appendix C.2) which is essential before transmission over the Apollo Token Ring and also reduces the required amount of disk buffering. The data are recorded in 20-Mbyte disk files containing ~ 2000 events each.

The CPU power of the filter is sufficient to handle an event rate of ~ 10 Hz, being limited by the data compression.

4.4 The event reconstruction system

Event reconstruction is the process by which raw subdetector data are converted into physical quantities such as particle energies or momenta. This task, which requires considerable computing resources, is traditionally done on mainframe computers, but the advent of high performance RISC-based workstations has made integration with the DAQ system economically feasible.

The main purpose of OPAL's on-line event reconstruction system is to provide reconstructed data of sufficient quality to begin physics analysis and to provide a quick feedback to the shift crew concerning the operation of the detector.

The large buffer of event files in the filter allows the reconstruction system to operate asynchronously from data acquisition. This is necessary because the data reconstruction requires access to calibration data which become available only after a delay of typically 30 min (sect. 5.5). More

buffering can reduce costs because the luminosity delivered by LEP varies significantly over a period of several hours. The processing power requirements can thus be reduced to correspond to average rather than peak event rates. Finally, buffering also renders the data acquisition insensitive to possible short failures of the reconstruction system. The current buffer capacity can accommodate 12 h of data acquisition at nominal LEP luminosity.

The reconstruction system is based on a network of HP Apollo workstations. The event reconstruction is done on three DN10000 workstations each hosting four CPU boards which share global memory and I/O resources. Nominal performance is 17 SPECmarks [32] per CPU board. Several less powerful workstations are used for data distribution and administration.

Full event reconstruction requires 26 CPU seconds for the average hadronic Z^0 decay and 2 CPU seconds for other events on a DN10000 processor. The event throughput of the system depends on the ratio of hadronic to non-hadronic events. In 1991 this ratio averaged 1:15, corresponding to a maximum throughput of 4 events/s. As all buffering, event handling and control is local to each DN10000, the system can be scaled by adding new machines.

All processing is done on a “per file” basis. The ORACLE database system (Appendix C.3.2) is used to maintain a history of each data file in the system and to store information on the number and types of reconstructed events. As new event files appear at the filter they are copied to a staging disk which serves as input to the reconstruction system. From here the files are distributed to the DN10000 workstations for event reconstruction as soon as the corresponding calibration files are available. Independently, the files are also copied to 630-Mbyte rewritable optical disks for permanent storage. The use of optical disks (rather than tapes) for data archiving has the advantage of preserving random access to events for future passes through the reconstruction process. The optical disk library [33] can store 32 removable optical disks. This provides operator-free random access to 20.8 Gbytes.

Each DN10000 runs four copies of the reconstruction program and processes one event file at a time. I/O latency for the reconstruction programs is minimized by use of asynchronous event handling tasks which copy events from the local disk file to shared memory according to a FIFO list of event pointers. After processing by the event reconstruction task, the events are copied from the shared memory to the local disk by event handling tasks. The events are subsequently ordered and merged into an output file corresponding to the original input file. Input and output data files are in compressed format (Appendix C.2.2) in order to minimize requirements on data transmission bandwidth and data storage capacity.

The output files of the reconstruction processes are copied to a large disk on a workstation which performs monitoring of the reconstructed events. This workstation also distributes the reconstructed event data using FTP [34] to three remote systems used for physics analysis (CERN IBM mainframe, OPAL off-line VAXcluster and UNIX-based workstation cluster [35]). The reconstructed events are available to OPAL physicists within a few hours of being recorded and the results of the monitoring program are available in the DAQ control room within about two hours.

5 GENERAL SYSTEMS AND SERVICES

There are several systems and services which, although not directly relevant to the flow of event data, are nevertheless important for the correct and reliable operation of the on-line system.

They fall into three broad categories: control, monitoring and calibration. The modularity of these services and the standardization of the interfaces with the “client” systems helps to preserve the flexibility of the on-line system as a whole.

The following sections describe the DAQ and slow-control systems, communication with the LEP control system, general services for monitoring detector performance, DAQ performance and data quality, and aspects of the calibration system of particular relevance to the on-line system. The underlying infrastructure used in the implementation of these services is described in Appendix C.

5.1 The DAQ control system

The DAQ control system runs on the on-line VAXcluster and performs the following functions:

- Routine control of the DAQ subsystems.
- Recovery procedures in case of occasional hardware or software malfunction.
- Maintenance of internal and external databases containing configuration and run-time parameters.

The DAQ control system program suite comprises:

- A control “engine” to perform the above tasks.
- A communication process tailored to handle the incoming and outgoing network messages between the engine and the DAQ subsystems.
- A communication process to link the operator interaction program to the engine.
- A server to retrieve information from the on-line ORACLE data base.

The above four processes run on the same CPU. The CATS subroutine interface (Appendix C.1.4) is used to provide the program-to-program network communication between different CPUs within and outside the VAXcluster.

5.1.1 Operator interaction

Operator interaction is mediated by a program which converts a sequence of menu selections and other graphical interactions into commands which can be interpreted by a client program. The operator interaction program runs on a VAX workstation using the DECWindows [6] window manager. Generic primitives of menus, parameter panels, message boxes and colour-coded cell matrices (to display status information) are provided. Specific instances of these primitives are created and destroyed dynamically by the client; in this case, the DAQ control engine.

5.1.2 Finite-state machines

The modular, hierarchical style of the DAQ hardware and software naturally lends itself to being modelled by a collection of finite-state machines (FSMs). Each FSM represents the states

of a program, a group of programs, a subsystem or the complete system. Commands and their parameters are associated with transitions between FSM states. A path between the current state of an FSM and the requested goal state is dynamically synthesized by connecting allowed elemental transitions until the first solution which connects the initial state to the goal state is found. The resulting path may consist of several elemental transitions connecting intermediate states.

The number of combinations of all FSM states in the complete system can be very large but the states are amenable to simplification by defining “manager” FSMs which have states mapping onto particular combinations of states of subordinate FSMs. Manager FSMs can in turn be subordinate to other managers, such that a hierarchy results with one manager FSM at the root. The states of this FSM provide the basis for global transitions of the system. An operator, in choosing to change the state of the root FSM, invokes a cascade of transitions throughout the hierarchy. In the current implementation there are ~ 80 FSMs defined which are organized hierarchically in, typically, four management layers.

5.1.3 State transition mechanism

State changes in the global system are brought about according to the following prescription and are illustrated in fig. 5 for manager FSMs and FSMs corresponding to external subsystems:

- (a) A path is sought from the current state to the desired goal state of an FSM.
- (b) A transition between the current state and the next state along the path is initiated. This results in one of two possibilities depending on whether the FSM is a manager FSM (M) or represents an external object (E), respectively:
 - (M) transitions of subordinate FSMs may be requested;
 - (E) commands may be sent to the external object after substitution of command parameters.
- (c) Completion of a transition also depends on the type of FSM, respectively:
 - (M) the transition is completed when all the subordinate FSMs reach the requested state;
 - (E) the new state is reported by the external object to DAQ control via a network message after successful execution of the commands. In both cases the next step along the path is attempted, or if the goal state has been reached, path completion is reported to the requesting manager FSM.

A typical example of a state transition occurs when the operator requests the start of a new data-taking run. In this case, a transition path from the current state of the root manager to the “running” state is sought. The root manager or “global” FSM has two subordinates which manage the subdetectors and central DAQ subsystems. The first step of the global path induces a transition from the subdetectors’ current state to the running state and a subsequent step induces the central components also to move to the running state. All subordinate FSMs must report completion of a step before the next step of the manager FSM is started. Thus, the desired synchronization between the subordinate FSMs is achieved. Similarly, the induced transition of the subdetectors’ manager FSM to the “running” state induces the subordinate FSMs (one per subdetector) to move to a “running” state and each of these transitions induces similar transitions in their subordinate FSMs. These FSMs, at the bottom of the hierarchy, correspond to external DAQ subsystems and their transitions induce run start commands to be sent to the corresponding subsystem. Successful

completion of the commands, associated with the transition, results in a change to a “running” state which is then reported to DAQ control by the external subsystem thereby completing the requested transition.

Errors may be reported by external objects during or independently of state transitions. In addition to the error code, the error report can also signal a change of state of a component. An error-specific transition can be automatically invoked when the error is reported. An optional configurable delay is used to ensure that all relevant error reports have been received before the transition is started. Error recovery then proceeds in the same way as state transitions requested by the operator.

5.1.4 Performance

To start a data-taking run takes one to two minutes, being dominated by the time taken by external components to execute the associated commands. Occasional failures by a subdetector to release the trigger busy signal or to deliver a subevent to the event builder are notified by the trigger or event builder which generate an error condition. Most such cases are recovered by DAQ control within several seconds without the need for operator intervention.

The DAQ control engine is currently implemented using the Prolog language [36] and using object-oriented programming techniques. However, a new version of the engine written in C++ language is under development and will increase the execution speed; allow easier interface with external software; benefit from the more advanced and more rigorously implemented object-oriented techniques inherent in C++; and provide a more easily maintained and portable program.

5.2 The slow-control system

The slow-control system [37] operates continuously and is independent of data acquisition. It is implemented on seven dedicated VME stations located in different electronics rooms, and one supervising station in the control room. The latter serves also as the main operator interface. Since safety-related parameters are also monitored, care has been taken to make the local operation of each station independent of external services like power distribution or the computer network.

Each station monitors typically 500 analog and 300 digital channels representing gas flow, high voltage, power supply status, temperature, etc. In addition, the slow-control software also runs in dedicated subdetector VME systems or in the LSCs of subdetectors with high-voltage equipment connected directly to the LSC. All communication between slow-control stations and the LSCs is based on OS-9Net (Appendix B.2) over Ethernet. One of the slow-control stations is connected to the event builder in order to include the data in the physics event record.

All monitored signals are continuously compared with nominal values and logged at regular time intervals. The operator is notified and automatic corrective actions may be taken whenever a value is outside a predefined range. All operating parameters like nominal values, tolerances and actions are software configurable. A menu-driven interface is provided for interactions locally and from other computers via remote log-in. A graphical operator interface on a Macintosh-II computer is used at the central station in the control room where the overall status of the detector is also displayed graphically.

5.3 Communication with the LEP control system

The OPAL on-line systems need to be able to exchange information with those of LEP for several reasons: LEP requires an automatic monitor of the beam conditions near OPAL in order to correctly adjust collimators and make beam orbit corrections, and OPAL requires data from LEP for monitoring purposes and for subsequent data analysis.

The LEP control computers are connected to the OPAL on-line system by the CERN-wide Ethernet network. A VAX workstation in the on-line VAXcluster exchanges information with the LEP computers using TCP/IP and stores the information from LEP in the network data base (NDB) (Appendix C.3.3) from where it is accessible to all VAX, Apollo and VME systems.

5.4 The monitoring systems

Most monitoring uses histograms filled by the various subsystems as the source of information. On-line and off-line histogram viewing use the same tools so that physicists need only be familiar with one utility.

Dynamically changing, run-time status information is monitored using a program capable of simultaneously driving many display screens and by a program which graphically represents the event flow through the event builder. Also, an event display program provides an immediate visualization of all or selected physics events.

5.4.1 Histograms

On-line histograms are produced by monitoring programs at different levels of the data acquisition. The LSCs perform detailed monitoring of subdetectors, the trigger LSC monitors trigger and dead-time conditions, the filter monitors the combined detector performance and event filtering, and the event reconstruction system monitors the combined detector performance and the quality of the resulting physical quantities.

All these different systems use the HBOOK histogram package [38]. For efficiency reasons, the histogram filling in the OS-9 based systems is normally not done in HBOOK but in simple arrays. These arrays are then periodically copied into an HBOOK structure. HBOOK histograms can be viewed from any workstation within the CERN-wide local area network using PAW [39] which accesses memory or disk-resident histograms in remote computers over TCP/IP. PAW users can, with the help of predefined panels, select viewing of histograms from any part of the system without prior knowledge of where the histograms are accumulated.

The central trigger system, the filter, and some subdetectors use Macintosh-II computers for local display of the histograms. The Macintosh-II computers are connected to VMEbus via VICbus, which provides an efficient memory-mapped access to the histograms. The trigger and filter histograms are permanently displayed on Macintosh-II computers in the control room. A Macintosh style user interface [40] allows easy selection of histograms, and the large, high-resolution colour screen permits the display of many histograms simultaneously.

5.4.2 Status display programs

A general-purpose status display program is used to simultaneously manage many pages of status information on simple terminal devices. The program extracts named information from NDB (Appendix C.3.3) and formats the display screen according to the description contained in configuration files. Items of special significance may be highlighted according to the capabilities of the display device.

The program periodically refreshes the displayed values of the items as stored in NDB and may also periodically step through a sequence of different pages of information on a single screen. Both page and screen refresh periods are configurable. The sources of information currently include the subdetector and central DAQ systems and the VAX workstation which communicate with the LEP control system.

An additional, more specialized status display program is used for the visualization of the event flow through the event builder. It provides important diagnostic information in case of DAQ errors. The program displays the progress of each event from each of the OPAL subdetectors through the sequence of steps that comprise the event building process. The information is stored by the event builder in NDB. The display program extracts the information and displays the status of all stored events as columns of a matrix, one column of events per subdetector, using colour codes to indicate the processing step for each event.

5.4.3 Message display

Messages managed by the EMU [41] system (Appendix C.4) may be displayed on simple terminals connected, for example, to the LSCs for local monitoring. However, the preferred way of viewing messages during common data acquisition is with a message display program which uses the DECWindows graphical interface to create a scrolling message buffer in which messages of special significance can be colour-coded according to an EMU message property.

5.4.4 Event display

An event display program is used for visual inspection of event properties and for immediate visual monitoring of the detector performance. Selective display of different categories of events is possible using the filter event classification. An example of a typical graphical representation of the raw data from some of the subdetectors is shown in fig. 6.

The program is written in FORTRAN [42] using Macintosh toolbox calls for fast graphical displays and runs in the MacSys [43] programming environment on a Macintosh-II computer. The Macintosh NuBus is connected to the event builder VICbus providing memory-mapped access to the event data structures. A 12-inch bitmap graphics display device (1024×768 pixels with a 256 colour palette) is used for the graphical output.

5.5 The calibration database system

The calibration database is used to maintain calibration data for the event reconstruction program. Many calibration values vary with time and correct values for any period must be available for reconstruction of events throughout the lifetime of the experiment.

Calibration data measured before or during data acquisition are often refined during physics analysis. Updates to the calibration database must therefore be accepted from both on-line and off-line computer systems. On-line event reconstruction must wait for the results of calibration performed during data acquisition to become available. This synchronization is also handled by the calibration database system.

The calibration database uses an OPAL-developed hierarchical database, OPCAL (sect. C.3.1). The master OPCAL database resides on a dedicated workstation on the Apollo network. Access and retrieval is done over TCP/IP. All calibration updates are merged into this database by a single “update server” process, in order to preserve the integrity of the database. The on-line event reconstruction system and the LSCs access the update server via special processes which also handle the synchronization of calibration and event data.

Critical calibration data measured during data acquisition by a few subdetectors are synchronized with the event data by means of a key value added to both the calibration data and the raw event data. The filter produces a list of the required keys for each recorded event data file. The event reconstruction process submits this list of keys when requesting calibration values for a given file. The calibration database returns the calibration data when data corresponding to all the file’s key values are available in the master calibration database and only then is the event reconstruction allowed to proceed.

The size of a set of calibration data corresponding to an event file is ~ 2.5 Mbytes. The total 1991 database amounts to 17 Mbytes.

6 COMMENTARY

The on-line system is described in this paper as it was used during the 1991 LEP physics run. It has considerably evolved since LEP start-up and will continue to do so. This is easily possible because of its hardware and software modularity which was one of the key points in the design. It enabled not only the replacement of modules without perturbing the rest of the system, but also the efficient distribution of the work to people.

6.1 Design experience

Computer-Aided Software Engineering (CASE) techniques were not generally used in the design of this on-line system. In the early design phase structured analysis diagrams [44] were used as a communication aid within the design group. The modularization of the system, with each “module” being manageable by one or two persons, has probably contributed to the success of the system in spite of not using CASE tools. The early designers had considerable experience with smaller systems and had a good idea of how to modularize the system based on real world experience. The modularization reflected the organization of the manpower: it was not hierarchical but, rather, autonomous, made of cooperating and communicating units.

6.2 Operational experience

The experiment is operated routinely with only three people on shift: one for data-quality monitoring and coordination with the LEP operators; one for data acquisition; and one to monitor

the detector hardware and safety aspects. This is possible because the comprehensive monitoring and the simple recovery procedures enable the people on shift to quickly localize, diagnose and correct most of the problems which may arise. In some cases the need for operator intervention is completely avoided: some DAQ time-out conditions are automatically handled by the DAQ control system and automatic emergency actions may be taken by the slow-control system. In some other cases the problem, once localized, can be expeditiously solved by the person on shift by resetting the appropriate system: an operation which, by design, does not disturb any other components when done in a controlled way. The need for a continuous presence of people with specialized expertise is therefore avoided.

The high quality of the data recorded for use in physics analyses is due to the extensive monitoring of the data at all stages of processing including the monitoring of fully-reconstructed events which are available within ~ 2 h. As a result, a negligible fraction of events were subsequently classified as unusable for any physics analyses.

6.3 Performance

During the 7-month running period in 1991, about four million events were recorded, containing 352 400 multihadron events. The overall DAQ efficiency was $\sim 91\%$ of which $\sim 3.5\%$ of the inefficiency was due to readout dead time. The time between unscheduled operator actions was typically several hours. Additional performance figures are given in table 4.

6.4 Upgrade plans

The readout dead time per event will be reduced to keep the dead time low at higher LEP luminosities. The jet chamber subdetector will introduce a transputer-based FADC readout. This will reduce its dead time to a constant value, independent of the number of hits in the chamber. A different approach will be taken to reduce the dead time of the z-chamber subdetector where double buffering will be implemented in the FADC readout. The vertex chamber subdetector will replace its ten MC68000 CAMAC-based processors by ten MC68020 VME-based single-board computers each with a dedicated CBD8110 CAMAC branch driver [45] interfaced to its VSB.

An MC68040-based single-board computer [46] will be added to all the local system crates in order to reduce the dead time and increase the maximum throughput. The readout tasks will be split between the two CPUs. The currently used CPU will be dedicated to handling the trigger interrupts and the data readout into the first stage of the buffer manager. All other tasks, including the handling of network protocols and peripheral devices, will be transferred to the four times faster MC68040. The aim is to reduce the overall readout dead time to 7 ms and be able to handle twice the LEP design luminosity with a maximum event rate in excess of 15 Hz.

A pretrigger will be introduced to cope with the reduced decision time available when LEP starts operating with a time between bunch crossings of $11 \mu\text{s}$ corresponding to doubling the number of electron and positron bunches. The pretrigger will act as an extension of the existing central trigger system.

A new subdetector, a silicon-tungsten luminosity calorimeter will be added for 1993.

6.5 Conclusion

The LEP experiments are of considerable size and complexity and span almost two decades from their design to their end. During this time they may undergo major changes both in the physics questions they address and in the detector hardware. This has implications on both the functionality and performance of the DAQ system.

The modular implementation, achieved through use of the VME standard and of workstations, has so far enabled the OPAL data acquisition system to adapt to new demands as they developed. The upgrades needed for the higher luminosities expected in 1992 are well understood and can now take advantage of new faster processors in order to achieve the required performance more cost-effectively than previously possible. Another benefit of this gradual improvement is that new developments can profit from operating experience of the present system and thus more easily be targeted at solving the real problems.

Acknowledgements

We are very grateful for the help we received from CERN personnel of the ECP and CN divisions during the design, implementation and operation phases of the on-line system. This especially concerns technical support in the VME/OS-9 area, assistance in networking, both hardware and software, and the CERN-developed software packages MODEL, PAW and ZEBRA. The work of the technical personnel of the collaborating institutes and of the CERN PPE division was essential for providing the necessary infrastructure.

A The VMEbus family and interfaces

A.1 The VMEbus family

The VMEbus is a 32-bit address and data, non-multiplexed, asynchronous backplane bus, supporting multiple masters and several addressing modes. The basic specification [47] covers a single backplane with a maximum of 21 modules.

The VSB (VME Subsystem Bus [48]) is an additional backplane bus used in conjunction with VMEbus to allow up to six modules to share data in private clusters. In the high-energy physics environment VSB is sometimes also used for interfacing readout systems directly to a CPU (table 2).

VICbus (VME Inter-Crate bus) is an emerging standard for a multimaster cable bus intended primarily for the interconnection of VMEbus crates, but which may also be used to connect VMEbus systems to other standard buses. VICbus provides software-transparent connections between VMEbus crates, as well as high-speed data transfers between interfaces over cables of up to 100 m in length. The VICbus standard is currently under development by an ISO/IEC working group, and should become an international standard during the course of 1992.

In OPAL, the predecessor of the VICbus, the proprietary VMVbus [49], is used to interconnect VME crates and workstations (Apollo DN10000 and Macintosh-II computers). The VME module used (VIC8250 [50]) also incorporates a triple-ported buffer memory accessible from the VME, VSB and VICbus. Another module with less functionality (VIP [51]) was used for some of the VME crate interconnections for the LSCs.

A.2 Interfaces to readout electronics

Front-end electronics interface to VMEbus in one of the following ways:

- CAMAC is interfaced to VMEbus with a VME CAMAC branch driver (CBD 8210 [45]).
- Fastbus is interfaced via a cable implementation of the VSB (FVSBI [52]). The VSB connects to a Fastbus board which has on-board dual-ported memory and sequencing logic for fast readout of Fastbus modules.
- The DL300 FADCs [13] are also interfaced via a cable implementation of VSB. This is particularly important in a multi-CPU crate, because it allows all CPUs to read out in parallel without interference.
- Custom-built VMEbus modules performing bit-map, hit list or multiplexed analog readout are used by several detectors.
- Several of the high-voltage systems are controlled directly from a VME-based controller (CAEN [53]).

B The OS-9 environment

OS-9 is a small, low-overhead, UNIX-like, multitasking system from Microware Inc. [54]. Some of the features, which make OS-9 interesting for real-time applications in VME are:

- Modularity of the system itself and of the user code.
- Position independence and re-entrancy of all code.
- Data modules for named shared data areas.
- Ease of writing device drivers.
- Provision of UNIX-like interprocess communication with pipes, event flags and signals.
- Per process exception handlers.
- Lightweight process management and scheduling.
- Both time slicing and real-time priority pre-emptive scheduling.
- A UNIX-like file system.

B.1 Device drivers

OS-9 comes with drivers for standard devices but in the DAQ environment many non-standard devices must be used and it is therefore important in the development phase that new drivers can be easily written and tested. In OS-9 new device drivers can be loaded while the system is operating which greatly facilitates their testing. OS-9 also supports multilevel drivers. Multilevel driver support for Ethernet is available from a third-party supplier (SYAC [55]). This allows implementation of multiple protocol drivers for Ethernet. TCP/IP, ISO, OS-9Net [54] and BOOTP [56] are all implemented over the same low-level Ethernet driver.

Device drivers have been developed to treat interrupts from CAMAC, VICbus and local and global trigger units, and for networking over the VME backplane (sect. B.2). The drivers developed for the data acquisition system often act merely as converters of interrupts into event flags which can be used to schedule normal tasks.

B.2 OS-9Net

OS-9Net provides communication between OS-9 systems. It is well integrated into the operating system and allows I/O system calls to be executed transparently on a remote system. This allows remote file access, change of working directory to a remote directory, remote log-in, etc. OPAL uses OS-9Net via Ethernet, implemented as a driver on top of the low-level driver. There is one such network connection per crate.

In crates with multiple processors OS-9Net is also implemented as a network on the VME backplane. A second low-level driver has been written which emulates Ethernet packet traffic on the backplane using the processors' dual-ported RAM and associated interrupt features for a mailbox mechanism. The higher-level driver for OS-9Net is basically the same as for the Ethernet implementation. This scheme allows file access by, and remote log-in to, secondary processors in the crate which have no peripherals of their own connected. The most important application is for pipes which extend between processors, thus providing straightforward remote interprocess communication.

B.3 File system

The OS-9 systems needed a common file system to give all systems access to the common software. NFS [57] would provide the right functionality, but was not available for OS-9 at the start of OPAL. A network disk system [58] developed at CERN for the ALEPH experiment was chosen instead and has been adapted by OPAL and used successfully. However, now that OS-9 supports NFS, this will be introduced in 1992. Many of the OS-9 systems also have a local hard disk which is used for program development and temporary data storage. The hard disks were introduced to allow network-independent operations.

B.4 FORTRAN

A FORTRAN compiler was required for porting existing code such as CERN libraries and OPAL track reconstruction programs to OS-9. Also, in the early days of OPAL, FORTRAN was still the preferred programming language for most of the physicists developing the on-line system. The RTF [42] compiler was used for porting the CERN code without major modifications to the latter. The many real-time extensions in RTF also made it suitable for writing data-acquisition code.

B.5 Booting OS-9 systems

The EPROMs in the single-board computers are set up to contain a minimal OS-9 system, with drivers for hard disk and Ethernet, and a BOOTP client protocol. Systems can be booted from the hard disk or from the network. In normal operation a three-level boot sequence is used in order to minimize the time it takes to reset an OS-9 system:

BOOTP	is used to load a file of modules common to all systems and to define each system's logical address used for TCP/IP communication.
Coldstart	is used to copy system-dependent modules to memory. All executable modules used for data acquisition are loaded in this way.
Warmstart	OS-9 finds these modules in memory and needs only start the relevant processes.

The BOOTP and Coldstart modules are loaded in RAM at addresses which OS-9 regards as EPROM space. The definition of OS-9's memory configuration and the path to the start-up file is defined in a battery-backed-up RAM. This allows all CPUs to have identical EPROMs, but different memory configurations and start-up files: an essential flexibility in a VME environment with many CPUs.

C Infrastructure

This Appendix gives some additional details of some of the infrastructure used in the implementation of the on-line system.

C.1 The communication network

A reliable communication network is essential for the correct operation of this distributed on-line system. Care has been taken to minimize the dependence on external systems not essential for data acquisition.

C.1.1 The local area network

Two local area networks (LANs) are used within OPAL; Ethernet and the Apollo Token Ring. All the VAX and VME systems and some of the Apollo systems are connected to Ethernet and all the Apollo systems are connected to the Apollo Token Ring. The Ethernet, at the experiment, is connected via a bridge to the CERN-wide FDDI ring [59] which provides a transparent connection to all Ethernets at the CERN site. The AppleTalk network [7] is used to interconnect Macintosh computers.

C.1.2 TCP/IP

TCP/IP [60] is today's *de facto* standard, available on all computer systems. In OPAL, the manufacturers' implementation is used on Apollo and VME/OS-9 systems and the MultiNet [61] implementation is used on the VAX/VMS systems. The higher level protocols such as TELNET [60] and FTP [60] are now available on all OPAL systems. In the early days of OPAL, when TCP/IP was not well implemented on VME/OS-9 and VAX/VMS, other protocols were implemented and are still used (sect. B.3).

C.1.3 OSI

The international Open System Interconnect (OSI) standard [60], was available as a partial implementation, OSI/TP4 [60], on VAX/VMS [62] during the design of the DAQ system. OSI/TP4 was also available as a portable product [63], which was installed on OS-9 by CERN. This provided a tool for program-to-program communication between VAX/VMS and VME/OS-9 systems. However, a standard programming interface was not available, so OPAL decided to use a CERN-defined interface CATS (see below) for program-to-program communication.

C.1.4 CATS

CATS [64] is a CERN-defined programming interface for program-to-program communication over OSI/TP4. It has been implemented on top of OSI/TP4 on both VME/OS-9 and VAX/VMS

systems and also on top of DECNET on VAX/VMS systems. All program-to-program communication for DAQ services, such as DAQ control, EMU (sect. C.4) and NDB (sect. C.3.3), is based on CATS. When Apollos were introduced into the DAQ system in 1991, CATS was also implemented on top of TCP/IP's socket interface on VAX/VMS and Apollos as a means of integrating the Apollos into existing services.

C.2 Data format

A standard data format (ZEBRA [65]) is used in both on-line and off-line software. However, for efficient use of available network and storage media capacity, a special compressed data format is also used.

C.2.1 ZEBRA

ZEBRA is a CERN-developed library which supports dynamic creation and manipulation of data structures in a FORTRAN 77 environment. The package includes support for sequential and direct access input and output of these structures and is available on most platforms used in high-energy physics experiments.

All of OPAL's event reconstruction and data analysis is based on ZEBRA, so it was natural to use ZEBRA also in the on-line environment. ZEBRA is also used indirectly by all histogram applications because the HBOOK package uses ZEBRA internally. The output of each subdetector LSC is a ZEBRA-structured subevent already in the same format used by the reconstruction software. The event builder merges all the subdetector ZEBRA structures into the full ZEBRA structure for OPAL events.

In the OS-9 systems and filter, the processing of event data is usually not done with the ZEBRA library for reasons of efficiency and program size. A set of routines developed by OPAL supports the basic functionality to access and manipulate data in ZEBRA-compatible data structures using pointer variables in C or pointer-based COMMONs in FORTRAN extensions. In particular, data copying for input and output is avoided.

The jet chamber processing (sect. 3.3) uses the COBRA [66] library which is call-compatible with ZEBRA but provides features needed in multiprocessor and real-time applications. This allows direct migration of code developed and used in off-line programs into the on-line environment.

C.2.2 Compressed data

The ZEBRA structures are optimized for fast access to data from FORTRAN programs and not for compact data storage, e.g. all integers are stored in 32 bits regardless of the required precision. A compressed data format was defined [67] to minimize the requirement on data storage and network bandwidth while still retaining the full information. Event data are packed into this format before storage and unpacked into the original ZEBRA structure before reconstruction. Information about required precision and resolution and the OPAL event structure is coded into the purpose-built packing and unpacking routines. The packing reduces storage and bandwidth requirements by almost a factor five at a cost of 14% increase in CPU time for event reconstruction. The compressed data format reduces the network bandwidth requirements to a level where the Apollo

Token Ring will be able to transfer events from the filter to the reconstruction system at event rates corresponding to more than two times LEP design luminosity.

C.3 Databases

Three database systems are integrated into the DAQ system: OPCAL, a hierarchically structured database system based on the ZEBRA package, is used for detector calibration purposes; ORACLE, a commercial relational database management system is mainly used for bookkeeping functions; NDB, a distributed database system, is used for dynamically changing status information. The motivation for the three choices is described in the following sections.

C.3.1 OPCAL

OPCAL [68] is a package of subroutines and auxiliary programs designed to allow storage and retrieval of calibration data for the OPAL experiment. Such calibration data are used mainly inside FORTRAN programs which are based on the ZEBRA memory management system. OPCAL itself uses the ZEBRA RZ [65] package for direct data access.

OPCAL is not a relational database system but it organizes the data in a hierarchical manner. Such organization is appropriate for the storage of a historical sequence of sets of calibration data. It allows access to the current version of the required dataset or that which was current at any time in the past. A hierarchical structure also accommodates easily that OPAL is organized in a simple hierarchy of subdetectors.

C.3.2 ORACLE

ORACLE [69] is a commercially-available relational database management system. Numerous interfaces exist to query and archive data including a command-line interpreter which accepts SQL (Standard Query Language) commands and a FORTRAN or C-callable subroutine interface both of which are routinely used in OPAL.

In OPAL, ORACLE databases are used by the DAQ system and the event reconstruction system mainly for dataset bookkeeping, run summary archiving and archiving of LEP machine parameters and background conditions.

The DAQ database resides on the on-line VAXcluster and is accessible over CATS from the other OPAL subsystems using a FORTRAN or C-callable function. The request to update the database is formatted in the application program as a simple text string in SQL syntax and is sent to a server for processing. A similar subroutine exists to allow update requests to be sent from VAXs to the server process using VMS mailboxes, and the DAQ control system also has an interface to the database for both update and retrieval. The database is accessible for update and retrieval of information from remote computers using DECNET.

The event reconstruction database resides on one of the workstations on the Apollo Token Ring. All access to the database uses the standard ORACLE remote access mechanism based on the socket library.

C.3.3 Network database

The OPAL Network Data Base (NDB) [70] is a distributed database designed to allow named items to be stored and retrieved with low overhead. Its primary use is for frequently updated volatile data to be made available to all participating computers. For example, status information is entered by the subsystems and retrieved by the status display programs. In addition, LEP parameters such as beam energy and fill number are entered by the workstation which communicates with the LEP control system and is retrieved by most of the other subsystems.

NDB has been implemented in a portable way. It currently runs on VMS, OS-9 and Domain/OS operating systems and can be called from C or FORTRAN. The information within NDB is volatile — it is not stored on any mass storage device.

The key points that give NDB the desired high performance are:

- Within any single CPU the database resides in an area of shared memory. This gives local programs very rapid access.
- Local caching of remote databases is implemented for data retrieval to avoid excessive network access. If a locally-cached copy is still valid it will be used, otherwise a new copy of the entire remote database will be cached.
- There is no central item name directory — efficient retrieval requires that the name of the remote system be explicitly specified.

C.4 Message processing

The Error Message Utility (EMU) package [41] provides a mechanism for routing error and other messages between the VAX/VMS, OS-9 and Domain/OS systems. The extension to OS-9 and Domain/OS was developed by OPAL. EMU implements a pipeline: messages are injected from user applications at one end, and pass down the pipeline for processing. Error message processing proceeds in four stages: message injection, decoding, routing and destination processing.

Messages originating from OS-9 or Domain/OS systems are injected into a pipe on the local system by the application program and are transferred over Ethernet to one of two processes on the on-line VAXcluster. One process accepts messages over OSI from all OS-9 systems and the other accepts messages over TCP/IP from all Domain/OS systems. These processes then inject the messages into the standard EMU pipeline on the on-line VAXcluster.

The pipeline has been further extended by OPAL by the addition of a “relay” process. One or more of these processes can be declared as standard EMU destinations and their function is to forward selected messages to a remote system using CATS subroutine calls, thus allowing messages to pass transparently across operating system boundaries.

References

- [1] The OPAL Collaboration, OPAL Technical Proposal, CERN/LEPC Report 83-4 (1983).
- [2] The OPAL Collaboration, Nucl. Instr. and Meth. A305 (1991) 275.
- [3] M. Arignon et al., Nucl. Instr. and Meth. A313 (1992) 103.
- [4] D.G. Charlton et al., The on-line event filter of the OPAL experiment at LEP, to be submitted to Nucl. Instr. and Methods.
- [5] Apollo, Token Ring, and DN10000 are trademarks of Hewlett-Packard Co., Cupertino, CA, USA.
- [6] VAX, VMS, VAXcluster, MicroVAX and DECWindows are trademarks of Digital Equipment Corp., Maynard, MA, USA.
- [7] Macintosh, Macintosh II and AppleTalk are trademarks of Apple Computer Inc., Cupertino, CA, USA.
- [8] Esone Committee, CAMAC, updated specification, EUR8500 (1983).
- [9] IEEE standard Fastbus modular high-speed data acquisition and control system, ANSI/IEEE Std. 960 (1986).
- [10] Motorola Inc., Phoenix, AR, USA.
- [11] UNIX is a trademark of AT&T Bell Laboratories, USA.
- [12] International Standards Organization, Geneva, Switzerland, Ethernet ISO 8802/3.
- [13] P. v. Walter and G. Mildner, IEEE Trans. Nucl. Sci. 32 (1985) 626 (now commercially available as the DL300 FADC system, Dr. B. Struck, D-2000 Tangstedt, Hamburg, Germany).
- [14] B. Lofstedt, TPD — time projection digitizer, internal report CERN/EP/EL F682 (1989).
- [15] A.A. Carter et al., Nucl. Instr. and Meth. A286 (1990) 107.
- [16] F. Bourgeois et al., IEEE Trans. Nucl. Sci. 34 (1987) 240.
- [17] G. Schuler, 96-channel Fastbus charge integrating ADC CIAFB F683C, CERN/EF Internal Note (1988).
- [18] A. Beer, G. Critin and G. Schuler, Nucl. Instr. and Meth. A234 (1985) 294.
- [19] R.L. Chase and I. Corbett, IEEE Trans. Nucl. Sci. 32 (1985) 645.
- [20] N. Bingeors and M. Burns, SIROCCO IV, front-end readout processor for DELPHI microvertex, eds M. Budinich et al., proc. of the Int. Conf. on the Impact of Digital Microelectronics and Microprocessors on Particle Physics, Trieste, Italy (1988) 116.
- [21] MX5, successor to MX3, RAL Reports 87-063 and 89-009, Rutherford Appleton Laboratory, Oxfordshire, UK.
- [22] MVME147, 25 MHz MC68030, MC68882, 4-Mbyte DRAM from Motorola Inc., Phoenix, AR, USA.

- [23] FIC8230, 16 MHz MC68020, MC68881, 1-Mbyte SRAM, VSB and DMAC; FIC8231, 24 MHz MC68030, MC68882, 1-Mbyte SRAM, VSB and DMAC (from CES, CH-1213 Petit-Lancy, Switzerland).
- [24] DPM8241, 4-16 Mbyte dual-ported memory VME/VSB (from CES, CH-1213 Petit-Lancy, Switzerland).
- [25] From Exabyte Corp., Boulder, CO, USA.
- [26] P.A. Elcombe, OS-9 data acquisition skeleton, CERN On-line Note 461 from the OPAL Collaboration (1992).
- [27] P. Bock and G. Tysarczyk-Niemeyer, Nucl. Instr. and Meth. A283 (1989) 813.
- [28] M.A. Ellis and B. Stroustrup, The Annotated C++ Reference Manual, Addison-Wesley Publ. Co. (1990).
- [29] SysV is a trademark of AT&T Bell Laboratories, USA.
- [30] 4.3BSD is a trademark of the Regents of the University of California, USA.
- [31] C. Boissat and P. Vande Vyvre, The MODEL buffer manager User Guide, CERN/CN Division (a version of the buffer manager which has been ported to UNIX is used in this application).
- [32] One SPECmark is the processing power of a VAX-11/780 on the SPEC benchmark suite (SPEC is a trademark of the Standard Performance Evaluation Corp., c/o Franson & Hagerty, 181 Metro Drive, Suite 300, San Jose, CA, USA).
- [33] HP Series 6300 Model 20 GB/A, Rewritable optical disk library system with two 5.25-inch rewritable optical drives and capacity of 20.8 Gbytes on 32 removable disks, Hewlett-Packard Co., Cupertino, CA, USA.
- [34] RFC-959, DDN Network Information Center, Menlo Park, CA, USA.
- [35] J.-P. Baud et al., SHIFT, the scalable heterogeneous integrated facility for HEP computing, proc. of the Conf. on Computing in High-Energy Physics, Tsukuba, Japan (March 1991).
- [36] Robin P. Middleton, Control of a LEP experiment using a knowledge-based system, *proc. of the Int. Workshop on Software Engineering, Artificial Intelligence and Expert Systems for High-Energy and Nuclear Physics, CNRS Lyons, France (19-24 March 1990)*.
- [37] A.K. Amundsen et al., Nucl. Instr. and Meth. A293 (1990) 145.
- [38] R. Brun and D. Lienart, HBOOK User Guide, Entry Y250, CERN Program Library.
- [39] R. Brun, O. Couet, C. Vandoni and P. Zanmarini, PAW, physics analysis workstation, Entry Q121, CERN Program Library.
- [40] Advanced Software Concepts, 9551, Route de Saint Laurent du Var, F-06610 Gaude, France.
- [41] P.C. Burkimsher, The EMU User Guide, CERN/CN Division (this document describes the VAX/VMS implementation).
- [42] H. von der Schmitt, RTF/68K real time FORTRAN 77 for 68K processors, Reference Manual, Physikalisches Institut, Universität Heidelberg, Germany (May 1989).

- [43] S. Cittolin et al., MacUA1, Macintosh-based development system, the UA1 Collaboration, CERN/EP Technical Note 90-01 (1990).
- [44] E. Jourdan and L.L. Constantine, Structured Design, Yourdon Press, New York (1978).
- [45] CBD8210 and CBD8110, CAMAC branch driver, from CES, CH-1213 Petit-Lancy, Switzerland.
- [46] MVME167, 25 MHz MC68040, 4-Mbyte DRAM, from Motorola Inc., Phoenix, AR, USA.
- [47] VMEbus: a standard specification for a versatile backplane bus, IEEE 1014/D1.0.
- [48] IEC BUS 822 — Parallel subsystem bus for the IEC BUS 821 (VSB), International Electrotechnical Commission 47B (Central Office) 22.
- [49] VMVbus from CES, CH-1213 Petit-Lancy, Switzerland.
- [50] VIC 8250, VME crate interconnect, 512-kbyte triple-ported memory VME/VSB/VIC, from CES, CH-1213 Petit-Lancy, Switzerland.
- [51] P. Farthouat, The VME interconnect port (VIP), proc. of the IEEE NSS Symp., Washington, USA (1986).
- [52] M. Weymann, The FVSBI — Fastbus to VSB interface — short description, CERN/EP Internal Note (October 1989), (this interface is now commercially available as FVSBI 9210 from CES, CH-1213 Petit-Lancy, Switzerland).
- [53] Costruzioni Apparecchiature Elettroniche Nucleari SpA, Viareggio, Italy.
- [54] Operating system for the MC68000 processor family, from Microware Systems Corp., Des Moines, Iowa, USA.
- [55] SYAC, I-34126 Trieste, Italy.
- [56] RFC-1084, DDN Network Information Center, Menlo Park, CA, USA.
- [57] NFS is a trademark of Sun Microsystems Inc., Mountain View, CA, USA.
- [58] T. Charity, OS-9 network disk server, CERN/EP Internal Note 88-52 and DATAQ 88-9, from the ALEPH Collaboration (1988).
- [59] Fiber-distributed data interface, International Standards Organization, Geneva, Switzerland, ISO 9314 (1989).
- [60] W.R. Stevens, UNIX network programming, software series, Prentice Hall, Englewood Cliffs, NJ, USA (1990).
- [61] MultiNet is a trademark of SRI International and of TGV Inc., Santa Cruz, CA, USA.
- [62] VAX OSI Transport Service, Digital Equipment Corp., Maynard, MA, USA.
- [63] Marben Electronique, Paris, France.
- [64] T.J. Berners-Lee and G. Heiman, The CATS transport service calling sequence, CERN/CN Division (December 1986).

- [65] R. Brun, M. Goossens and J. Zoll, ZEBRA dynamic data structure and memory manager, Entry Q100, CERN/CN Program Library.
- [66] O. Schaile, COBRA, a memory manager for real time and multiprocess applications, CERN/CN On-line Note 465, the OPAL Collaboration (1992).
- [67] K. Riles and T.J. Smith, The dense data User's Guide, CERN/CN Off-line Note 486, the OPAL Collaboration (1992).
- [68] R. Cranfield, B. Holl and R.W.L. Jones, OPCAL User Guide — OC501, CERN/CN Off-line Note 476, the OPAL Collaboration (1990).
- [69] ORACLE, ORACLE Corp., 20 Davis Drive, Belmont, CA, USA.
- [70] P.A. Elcombe, Network data base — release notes, version 11, CERN/CN On-line Note 464, the OPAL Collaboration (1992).
- [71] P.P. Allport et al., The OPAL silicon microvertex detector, to be submitted to Nucl. Instr. and Methods.
- [72] J.R. Carter et al., Nucl. Instr. and Meth. A286 (1990) 99.
- [73] H.M. Fischer et al., Nucl. Instr. and Meth. A283 (1989) 492.
- [74] J.D. Hobbs et al., A modular electronics readout system for the OPAL electromagnetic pre-sampler, to be submitted to Nucl. Instr. and Methods.
- [75] C. Beard et al., Nucl. Instr. and Meth. A286 (1990) 117.
- [76] S. Dado et al., Nucl. Instr. and Meth. A252 (1986) 511.
- [77] M. Bramhall and S. Jaroslowski, IEEE Trans. Nucl. Sci. 36 (1989) 380.
- [78] A.A. Carter et al., Nucl. Instr. and Meth. A250 (1986) 503.

Table 1: Summary of subdetector types and number of detector channels. Also included are the trigger subsystems, TR and TT. The abbreviations indicate the DAQ subdetectors and are also used to identify the subdetectors in the subsequent tables.

	Subdetector		Type	Channels	Refs
Central tracking system	SI	Microvertex detector	Silicon microstrips	15725	[71]
	CV	Vertex chamber	Drift chamber	648	[72, 2]
	CJ	Jet chamber	Drift chamber	7680	[73, 2]
	CZ	Z chambers	Drift chamber	2304	[2]
Time-of-flight	TB	Time-of-flight detector	Scintillator	320	[2]
Electromagnetic calorimeter	EB	Barrel calorimeter	Lead glass calorimeter	9440	[2]
	PB	Barrel presampler	Limited streamer tubes	21504	[2, 74]
	EE	Endcap calorimeter	Lead glass calorimeter	2264	[2]
	PE	Endcap presampler	Thin, saturated gain wire chambers	6080	[75, 2]
Hadronic calorimeter	HT	Towers	Limited streamer tubes	1696	[2]
	HS	Strips		56146	[2]
	HP	Poletip towers	Thin, saturated gain wire chambers	336	[76, 2]
		Poletip strips		10240	
Muon detector	MB	Barrel	Drift chambers	1320	[2]
	ME	Endcap	Limited streamer tubes	42496	[2]
Forward detectors	FD	Tube chambers	Proportional streamer tubes	768	[2]
		Drift chambers	Drift chambers	192	
		Other components	Calorimeters and scintillators	216	
Trigger system	TR	Global trigger system			[3, 2]
	TT	Track trigger system			[77, 78, 2]

Table 2: Summary of subdetector digitizing electronics. The subdetector abbreviations are defined in table 1.

Subdetector	Readout channels	Digitizing electronics	Range (bit)	Digitized event size (bytes)	Digitizer crate	Interface to LSC
SI	15725	10-bit ADC SIROCCO IV [20]	10	31450	Fastbus	VSB
CV	648	Custom TDC [15]	12	≤ 18144	Custom, CAMAC	VME
CJ	7680	100 MHz 6-bit FADC DL300 [13]	8	$\sim 2\,500\,000$	Custom	VSB
CZ	1152	100 MHz 6-bit FADC DL300 [13]	8	$\sim 432\,000$	Custom	VSB
TB	320	ADC LRS-2280	12	640	CAMAC	VME
	320	TDC LRS-2228A	11	640	CAMAC	VME
EB	9440	ADC CIA-Fastbus [16]	15	37 760	Fastbus	VSB
EE	2264	ADC CIA-Fastbus [16]	15	9 056	Fastbus	VSB
PB	21504	Multiplexed ADC (32×21 fold)	15	86 016	VME	VME
PE	6080	Multiplexed ADC (32×24 fold)	12	21 160	VME	VME
HT	976	ADC CIA-CAMAC [18]	15	3 904	CAMAC	VME
	720	ADC LRS-2280	12	1 440	CAMAC	VME
HS	56146	Bit pattern unit	1	7 020	VME	VME
HP towers	336	ADC CIA-CAMAC [18]	15	1 344	CAMAC	VME
HP strips	10240	Bit pattern unit	1	1 280	VME	VME
MB	1320	12.5 MHz 8-bit FADC TPD [14]	8	$\sim 150\,000$	Fastbus	VSB
ME	42496	Multiplexed ADC (32×9 fold)	12	84 992	Custom, VME	VME
FD tube ch.	768	Multiplexed ADC (32×3 fold)	12	1 536	Custom	VME
FD drift ch.	192	50 MHz 6-bit FADC DL300 [13]	8	8 192	Custom	VSB
Other comp.	192	ADC CIA-CAMAC [18]	15	384	CAMAC	VME
Other comp.	24	ADC LRS-2249	10	48	CAMAC	VME
Other comp.	120	TDC LRS-2228A	11	240	CAMAC	VME
TT	2592	Custom [77]		$< 35\,000$	Custom, Fastbus	VME, VSB
TR	696	None	1	87	Custom	VME

Table 3: Summary of DAQ subdetector properties. The number and type of additional processors per main LSC processor [22,23] is given. The subdetector abbreviations are defined in table 1.

Subdetector	LSCs	Additional processors		Dead time	Event size	
		No.	Type		Hadron	Others
				[ms]	[kbyte]	[kbyte]
SI	1	14	MC56001 DSP	3	4.6	3.3
CV	1	10	MC68000	3	2.9	2.1
CJ	1	20	MC68040	6–15	141.1	30.4
CZ	1	6	MC68020	6–12	18.4	3.4
TB	1			4	1.1	0.2
EB	2			3	2.8	2.0
EE	2			3	5.7	4.9
PB	1	4	MC68020	7	8.2	5.2
PE	2			6	0.5	0.2
HT	2			6	2.3	2.1
HS	2			3	1.4	0.7
HP	2			9	0.5	0.5
MB	2			5	1.8	1.7
ME	1	3	ADSP-2100A	8	0.8	1.0
		3	MC68000			
		3	MC68030			
FD	2	1	MC68020	9	0.4	0.5
TT	1			Variable	5.3	0.9
TR	1				0.2	0.2
Total					211.0	60.3

Table 4: Typical values for the 1991 running period.

LEP bunch-crossing rate	45 kHz
LEP luminosity	$0.8\text{--}0.4 \times 10^{31} \text{ cm}^{-2} \text{ s}^{-1}$
Trigger rate	4.0–2.5 Hz
Readout dead time	15 ms
Fraction of events marked by filter for rejection	35%
Event size: multihadron (other)	211 (60) kbyte
Compressed event size: multihadron (other)	46 (12) kbyte
Filter input (output) data transfer rate at 4 Hz	$270(41) \text{ kbytes s}^{-1}$
Number of optical disks recorded in 1991	140

Figure 1: The OPAL detector. The figure illustrates the position and scale of the subdetectors

Figure 2: The figure shows schematically the major components of the DAQ system and the principal connections between them.

Figure 3: A generic LSC for a split subdetector shown with VICbus interface to both event builder and second LSC. VME and VSB interfaces to read out electronics are shown.

Figure 4: Layout of the event builder with VICbus connections to local system crates and filter. VSB is used as an internal local bus. The processors are all of type FIC8230 with on-board VSBinterface and DMA.

Figure 5: Illustration of the finite state machine (FSM) state transition mechanism. The top part of the figure illustrates the FSM hierarchy. Manager type FSMs are labelled "M" and FSMs corresponding to external DAQ subsystems are labelled "E". The bottom left hand side of the figure corresponds to a manager FSM and illustrates the relationship between manager and subordinate FSM's. All subordinate FSM's must have reached the expected state in order that the manager FSM can reach its next state. The bottom right-hand part of the figure corresponds to an FSM which represents an external DAQ component. In this case, a transition from the current state to the next state induces commands to be sent to the corresponding external subsystem and completion of the transition must be reported by the external subsystem before the next transition can proceed.

Figure 6: An example of a typical event graphically rendered by the event display program. Calorimeter energies have been summed and displayed as polygons with size proportional to energy. Hits in the tracking chambers are displayed as points and jet chamber tracks reconstructed by the CJ front-end processors are displayed as curved lines. Hits in the time-of-flight detector are shown as small rectangles between the jet chamber and calorimeters and hits in the outer muon detectors are shown as short line segments.