

CERN-LHCC-2005-018
ALICE TDR 012
15 June 2005

ALICE

Technical Design Report

of the

Computing

Cover design by CERN Desktop Publishing Service.
Visualisation of ALICE Experiment as simulated by AliRoot.

Printed at CERN
June 2005.

ISBN 92-9083-247-9

ALICE Collaboration

Contents

1	Basic parameters and raw data structure	1
1.1	Introduction	1
1.2	Raw data structure	1
1.2.1	Trigger, Data Acquisition, and High-Level Trigger systems	1
1.2.2	Raw data format	2
1.2.3	Common data format	3
1.2.4	Central Trigger Processor readout and interaction-record data format	5
1.2.5	High-Level Trigger readout	7
1.2.6	The DATE ROOT recorder	8
1.3	Computing data challenges	8
1.3.1	Introduction	8
1.3.2	Performances and results	10
1.3.3	Future plans	12
	References	13

1 Basic parameters and raw data structure

1.1 Introduction

This chapter describes the basic parameters of the ALICE data, used as input to the computing model and resources estimate. A brief description of the ALICE Data Acquisition (DAQ) system is given, together with a discussion on the Computing Data Challenges.

1.2 Raw data structure

1.2.1 Trigger, Data Acquisition, and High-Level Trigger systems

The architecture of the ALICE Data Acquisition and its interfaces with the Trigger and the High-Level Trigger are illustrated in Fig. 1.1 and detailed in the ALICE Technical Design Report on Trigger, Data Acquisition, High Level Trigger, and Control System [1].

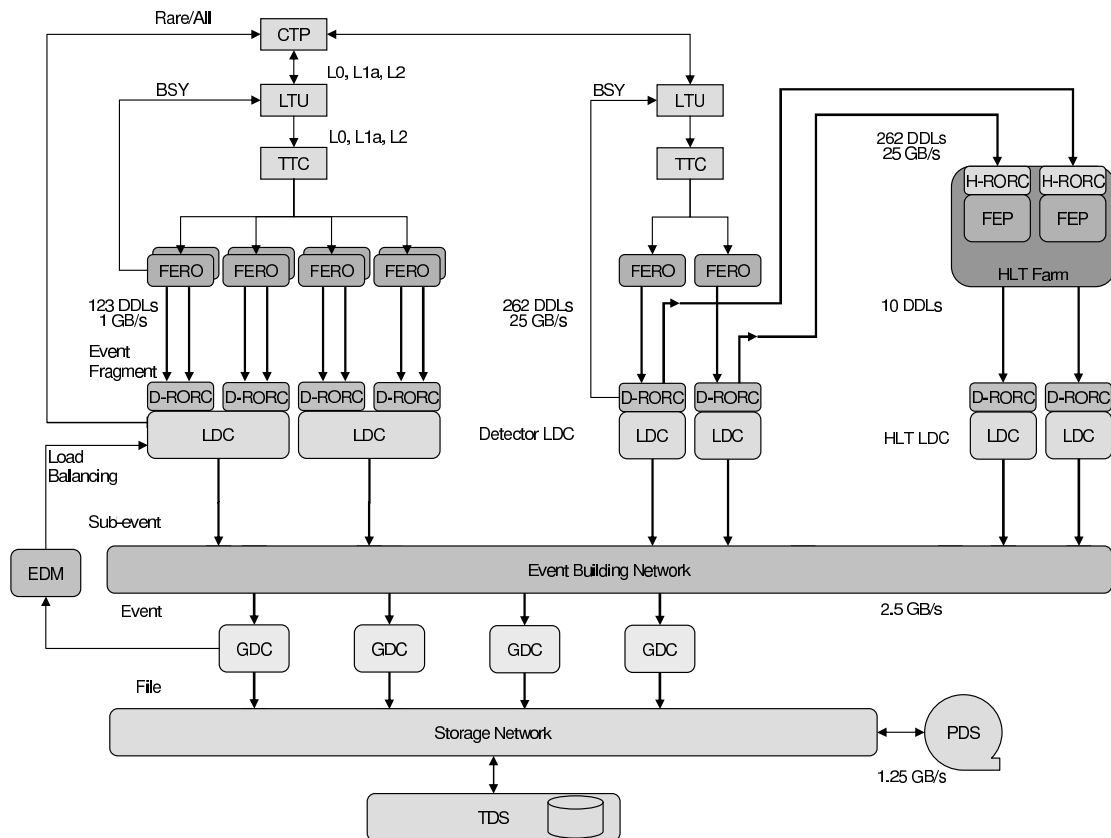


Figure 1.1: DAQ architecture overview.

The detectors receive the trigger signals and the associated information from the Central Trigger Processor (CTP) [2], through a dedicated Local Trigger Unit (LTU) [3] interfaced to a Timing, Trigger and Control (TTC) system [4]. The Front End Read Out (FERO) electronics of the detectors is interfaced to the ALICE Detector Data Links (DDLs). The data produced by the detectors (event fragments) are injected into the DDLs using a common protocol.

At the receiving side of the DDLs there are PCI-based electronic modules, called DAQ Readout Receiver Cards (D-RORCs). The D-RORCs are hosted by the front-end machines (commodity PCs), called Local Data Concentrators (LDCs). Each LDC can handle one or more D-RORCs. The event fragments originated by the various D-RORCs are logically assembled into sub-events in the LDCs.

The CTP receives a busy signal from each detector. This signal can be generated either in the detector electronics or from all the D-RORCs of a detector. The CTP also receives a signal from the DAQ enabling or disabling the most common triggers. It is used to increase the acceptance of rare triggers by reducing the detector dead-time. This signal is function of the buffer occupancy in all the LDCs.

The role of the LDCs is to ship the sub-events to a farm of machines (also commodity PCs) called Global Data Collectors (GDCs), where the whole event is built (from all the sub-events pertaining to the same trigger). The GDCs also feed the Transient Data Storage (TDS) with the events that eventually end up in Permanent Data Storage (PDS). The PDS is managed by the CERN Advanced Storage Manager (CASTOR) [5].

All these hardware elements are driven and controlled by the Data Acquisition and Test Environment (DATE) software framework [6] developed by the ALICE DAQ project. The coherence of the whole project is ensured by this common software framework composed of different layers of modules. A bottom layer includes the memory handling, the process synchronization, and the communication modules. The application layer includes the data-flow applications (detector readout, event building, and data recording). DATE has been used since several years by many ALICE test beams. Most of the ALICE detectors have already realised a complete integration of their readout system with the DDL and the DATE software.

The High Level Trigger (HLT) system [1] receives a copy of all the raw data. The data and decisions generated by HLT are transferred to dedicated LDCs.

1.2.2 Raw data format

The ALICE raw data format is formulated according to the needs of the computing model and following the architecture of the Trigger, DAQ and HLT systems [1].

The raw data format covers the following pieces of data:

- The event fragment is the block of data as sent by the detector electronics over the ALICE DDL. A common data format is defined [7] for all the data blocks transferred by every detector readout electronics over the DDL [8, 9]. This data format contains all the information needed by the DAQ for the sub-event and event building.

The data format described here is generated by the readout electronics. It covers the mandatory minimum data format for all the data blocks sent over the DDL and provides the information required to identify the data in the DAQ system. The identification of a data block and its processing is based on the format version, the event identification and the trigger information. The corresponding fields of the common data format header are therefore mandatory.

In addition, three optional fields are added to the common data format header: the block length, the event attributes, and the Region Of Interest (ROI) data. Extensions could be defined in the data themselves.

The format used for the data generated by the CTP is also presented hereafter.

- The sub-event is the block of data resulting from the merging of one or several event fragments. It is generated by the LDC and is the smallest amount of data that can be monitored in the DAQ system.
- The event is the assembly of all the sub-events pertaining to the same physics interaction. It is generated by the GDC.

1.2.3 Common data format

Data transferred from the front-end electronics to the LDCs is formatted as follows (Fig. 1.2):

- A header describing the associated data block(s), the trigger conditions, the error and status conditions and other information dependent on the readout electronics.
- Zero, one, or more data blocks belonging to the same physics or software trigger may follow the header. The data blocks are optional and may be skipped if there is no valid data associated with a given trigger.

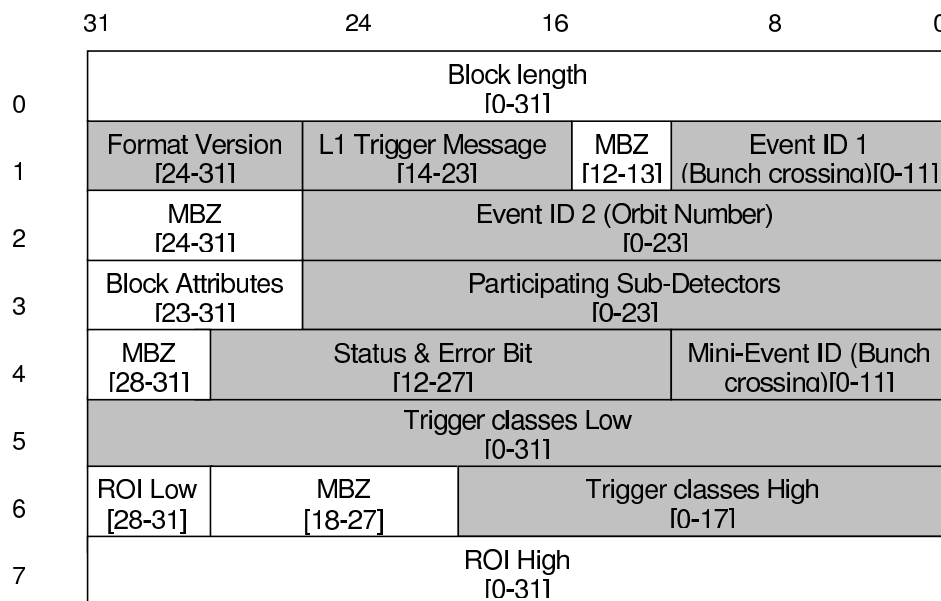


Figure 1.2: Common data format header.

The various fields of the common data format header are either loaded using the data transmitted by the ALICE Trigger system or created locally by the front-end electronics when running without the ALICE Trigger system (e.g. for standalone tests). The presence or absence of the ALICE Trigger system is marked by the trigger information unavailable status bit. When running without the ALICE Trigger system, the information contained in most of the fields is irrelevant.

The fields of Fig. 1.2 marked with a white background (e.g. Block length) are optional and can be left void if their value is not available or not needed. The fields marked in a grey background (e.g. Format Version) are mandatory and are filled with their correct value, as defined at the moment the event is generated. All fields marked MBZ (Must Be Zero) are reserved for future use and are set to zero by the readout electronics. A description of the individual fields of the common data format header follows.

Block Length

The Block Length is an optional field. It can be filled in by the detector readout electronics to indicate the total length of the data block including header and payload(s). The length is expressed in bytes being transferred over the DDL.

Format Version

The Format Version indicates which version of the present data format is used. The presence of this field will provide backward compatibility in case of change or upgrade.

L1 Trigger Message

The L1 Trigger Message consists of selected parts of the L1 trigger word [2]. This information is distributed over the TTC [4] to the detector readout. This field is a direct (bit-by-bit) copy of the first ten data bits of the L1 Data Message Word 1. It contains the information given in Table 1.1.

Table 1.1: L1 Trigger Message fields.

L1 Trigger Message field of DDL header	Data payload of the L1 Data Message Word 1
Bit 23–20	Spare bits
Bit 20	CIT bit
Bit 19–16	RoC[4...1]
Bit 15	ESR bit
Bit 14	L1SwC bit

Event ID

The LHC clock supplies the event identification in ALICE. This clock is distributed to all the detector readout units by the TTC system used as trigger distribution network. The current LHC design foresees 3564 bunches in one orbit. The LHC clock identifies each bunch crossing within an orbit and signals the beginning of a new orbit. Currently the TTC foresees 12 bits for the bunch-crossing number. The Trigger system includes a cyclic counter of 24 bits to count the orbit. This scheme uniquely identifies every bunch crossing in a period of more than 20 minutes ($(2^{24}-1) \times 88 \mu\text{s} = 1476 \text{ s} = 24 \text{ min}$), which is sufficient for this purpose. Further identification is added by the DAQ to uniquely identify one event in a run. The information stored in the Event ID fields (1 and 2) is transmitted by the CTP. It is distributed over the TTC in a dedicated message part of the L2 accept message and received by the detector readout electronics through the TTC Rx chips. When running without the ALICE Trigger system, the Event ID 1 field is set to zero and the Event ID 2 contains an incremental, unsigned binary number, to be reset at front-end electronics reset.

Block Attributes

The Block Attributes is an optional field that can be freely used by the detector groups to encode specific information.

Participating Sub-Detectors

The mask of participating detectors is a mandatory field. Its information is produced by the CTP only while handling software triggers. It is distributed over the TTC in a dedicated message part of the L2 accept message and received by the TTC Rx chips. The received values are copied as is in the Participating Sub-Detectors field.

Status & Error Bit

This is a mandatory field, to be loaded by the readout electronics under all running conditions. An error or status condition occurring before, during, or right after the front-end electronics readout is signalled by setting to one the corresponding bit(s) of this field. The assignment of the individual bits is described in Table 1.2.

Table 1.2: Status and error bits definitions.

Bit 12	Trigger overlap error: (L1 received while processing another L1)
Bit 13	Trigger missing error: (L1 received while no L0 received, or L2a or L2r received while no L1 received)
Bit 14	Data parity error
Bit 16	Control parity error (instruction and/or address)
Bit 17	Trigger information unavailable
Bit 18	Front-end electronics error
Bits 19 to 27	Reserved for future use

Mini-Event ID

The Mini-Event ID is the value of the bunch-crossing counter of the local TTC Rx chip at the time of the detector readout. It is a mandatory field. When running without the ALICE Trigger system, the Mini-Event ID field is set to zero.

A local event identification is also included in the common data format for cross-verification with the global event identification. This local event identification identifies the local value of the LHC clock at the time the detector has received the L1 trigger signal. It is based on the event identification reduced to the bunch crossing only, as counted by the TTC Rx chip [10]. The local bunch-crossing counters of all the TTC Rx chips of the experiment must be synchronous.

A key issue is to resynchronize them at regular intervals to ensure that this synchronism is maintained. The solution chosen is to use the mechanism foreseen by the TTC system. The local bunch-crossing counter in the TTC Rx chip can be automatically reset by a fast signal, synchronous with the LHC orbit. The LHC orbit signal is delivered by the TTCmi module [11]. This signal can then be sent over the TTC as a short-format broadcast signal. Proper usage and setting of the TTCvi module [12] guarantees that the TTC Rx chip receives this reset command by the end of the LHC extractor gap. The TTCvi includes four independently programmable timing signals. The bunch counter reset command uses the highest-priority programmable timing signal of the TTCvi.

Trigger classes (Low & High)

For physics triggers, the bits encoded in the Trigger classes' low and high fields are taken as is from the Trigger Level 2 accept message.

Region Of Interest (ROI) (Low & High)

The ROI data is distributed over the TTC system. The value—if available—should be stored in the ROI Low and ROI High fields. These fields are optional.

1.2.4 Central Trigger Processor readout and interaction-record data format

The CTP readout and the interaction-record data are generated by the CTP and transmitted to the DAQ via the DDL. The hardware and the communication procedure are standard—identical to the channels that transmit the sub-detector readout. The nature of the data, and the timing and rate of their generation, on the other hand, differ significantly from the sub-detector readout and are formatted by a customized data format. The CTP readout contributes to the event-building task. It is a redundant channel that carries exactly the same information broadcast to all the participating sub-detectors (L2a Message) at the time of L2a decision. It is used by the DAQ system to resolve error conditions.

The Interaction Record is an aid in pattern recognition. The generation of the record is continuous, rather than triggered by any CTP or DAQ action. Data do not interfere with any on-line operation—they only need to be archived for off-line use. Inside the DAQ, data from the CTP are formatted by the dedicated software to become compatible with the standard readout format used by the sub-detectors.

The CTP readout data format

For each L2a decision, the CTP transmits to the DAQ a data block containing the same information that is also broadcast to all the participating sub-detectors. The CTP readout block is uniquely marked by the Block Identifier bit cleared. The CTP readout block is shown in Fig. 1.3. The abbreviations used are summarized in Table 1.3.

Word	[31..14]	[13]	[12]	[11..0]
1				BCID [11..0]
2	<i>Don't care</i> (0)	BlockID = 0	<i>Don't care</i> (0)	OrbitID [23..12]
3				OrbitID [11..0]

<i>Physics trigger</i>		<i>Software trigger</i>			
Word 4	Bit	Data	Word 4	Bit	Data
	[31..14]	<i>Don't care</i> (0)		[31..14]	<i>Don't care</i> (0)
	[13]	BlockID = 0		[13]	BlockID = 0
	[12..11]	<i>Don't care</i> (0)		[12..11]	<i>Don't care</i> (0)
	[10]	ESR		[10]	<i>Don't care</i> (0)
	[9]	<i>Don't care</i> (0)		[9]	CIT
	[8]	L2SwC = 0		[8]	L2SwC = 1
	[7..2]	L2Cluster [6..1]		[7..2]	<i>Don't care</i> (0)
	[1..0]	L2Class [50..49]		[1..0]	<i>Don't care</i> (0)

Word	[31..14]	[13]	[12]	[11..0]
5		BlockID = 0		L2Class [48..37]
6	<i>Don't care</i> (0)		<i>Don't care</i> (0)	L2Class [36..25]
7				L2Class [24..13]
8				L2Class [12..1]

Word	[31..14]	[13]	[12]	[11..0]
5		BlockID = 0		L2Detector [24..13]
6	<i>Don't care</i> (0)		<i>Don't care</i> (0)	L2Detector [12..1]
7				<i>Don't care</i> (0)
8				<i>Don't care</i> (0)

Figure 1.3: CTP readout data format.

Table 1.3: Abbreviations used in the CTP readout data format.

BlockID	Block Identifier bit: cleared (0) for the CTP readout asserted (1) in case of the Interaction Record
BCID[11...0]	Bunch-crossing number, part of the Event Identifier
OrbitID[23...0]	Orbit number, part of the Event Identifier
ESR	Enable Segmented readout flag (RoI option)
L2SwC	Software Class L2 trigger status: cleared for the physics trigger; asserted for the software trigger
L2Cluster[6...1]	Cluster [6...1] L2 trigger status flag
L2Class[50...1]	Class [50...1] L2 trigger status flag
CIT	Calibration Trigger flag
L2Detector[24...1]	Detector [24...1] L2 trigger status flag

The CTP sends data blocks of a constant length of seven words to the DAQ. The first three words always contain the Event Identifier (bunch-crossing and orbit number of the corresponding event). The remaining words (Word 4–8) carry different information in the case of the physics trigger (L2SwC = 0) and in the case of the software trigger (L2SwC = 1).

The interaction-record data format

For every interaction detected, the CTP transmits to the DAQ the bunch-crossing number and the type of interaction (peripheral or semi-central). These informations are packed together into interaction-record data block. There is at least one interaction-record data block for each LHC orbit, even if no interaction has been detected during the orbit. The data format is shown in Fig. 1.4. The abbreviations used are summarized in Table 1.4.

The first two words of the data block contain the number of the corresponding LHC orbit (24 bits). They are followed by a string of words containing the numbers (12 bits) of the bunch-crossing at which the interactions have been detected and the corresponding Interaction-Type (InT) descriptors. One Interaction Record is sent to the DAQ every 250 interactions. During a run, the DAQ receives, in sequential order, the Interaction Record data blocks for all the LHC orbits.

Word	[31..14]	[13]	[12]	[11..0]
1	<i>Don't care (read 0)</i>	BlockID = 1 (Interaction Record)	Err	Orbit number [23..12]
2			Err	Orbit number [11..0]
3			InT	BC number [11..0]
4			InT	BC number [11..0]
⋮			⋮	⋮
n			InT	BC number [11..0]
⋮			⋮	⋮
251			InT	BC number [11..0]
252			InT	BC number [11..0]
253			0	<i>Incomplete record (hFFF)</i>
254			0	EOBTR (h0B4)

Figure 1.4: Interaction Record data block.

Table 1.4: Abbreviations used in the interaction readout data format.

BlockID	Block Identifier bit: clear (0) for the CTP readout set (1) in case of the Interaction Record
BCID[11..0]	Bunch-crossing number, part of the Event Identifier
Err	Transmission Error flag
InT	Interaction Type flag: cleared (0) for peripheral and asserted (1) for semi-central interaction

1.2.5 High-Level Trigger readout

The HLT system transfers three kinds of information to the DAQ system:

- The HLT decision tells the DAQ whether an event has to be read out or not. It can take different values: a rejection, a complete readout, or a partial readout. The HLT decision is implemented as a refined list of sub-events that have to be read out. This is a refinement of the decision taken by the Trigger system.

- The data modified by the HLT data compression. Again, here this will be implemented as a refined list of subevents: replacing a set of sub-events by another one containing the same data but in a processed form.
- New data produced by the HLT such as Event Summary Data (ESD).

1.2.6 The DATE ROOT recorder

The requirements for the DATE recorder are the following:

- It is part of the DAQ framework for the dataflow and control aspects.
- It writes the physics data in ROOT format and formats the data into objects.
- It uses CASTOR as Mass Storage System (MSS) and is interfaced to the Grid for the file catalogue and the metadata repository.

The DATE recorder has therefore the following characteristics:

- The recorder process is started and stopped by the DATE Run Control in synchrony with the rest of the DAQ system.
- It gets the data as a set of pointer and size to the different subevents which constitute one event and writes all the data pertaining to the same event into the same file.
- The DATE recorder archives the raw events as one or several streams of C++ ROOT objects in TDS. The data files are then archived to the CASTOR MSS.
- The ROOT objects are formatted by a library included in the offline software. These objects are formatted to allow an easy access and an efficient processing by the ALICE Offline reconstruction software. The objects include a set of physics raw data, and a set of ESD.
- The DATE recorder is also interfaced to the Grid-based run catalogue database that contains information on the location of the raw events, the description of each recording unit, and some run-related information (trigger parameters and run conditions).

1.3 Computing data challenges

1.3.1 Introduction

Since 1998, the ALICE team and the CERN/IT division have jointly executed several Computing Data Challenges (CDCs): cadenced, large-scale, high-throughput, and distributed-computing exercises, whose goals are to prototype the ALICE Data Acquisition and computing systems, to test hardware and software components, and to accomplish an early integration of the overall ALICE computing infrastructure.

The first CDC took place in 1998. Six CDCs have been held so far. Year after year, existing components have been replaced with new versions and new elements have been introduced in the chain. The exercise started by using material already deployed at CERN such as Test-Beam Data-Acquisition systems, Fast-Ethernet, and the Gigabit-Ethernet based backbone and by exercising the system with dummy data.

As the exercise evolved in time and requirements, the focus has shifted towards a continuous early-prototyping effort and to the deployment of dedicated, highly-available fabrics, avoiding interference with other CERN activities whenever possible. The data used during the CDC have evolved into realistic physics data simulated with the ALICE simulation program.

Goals of the Computing Data Challenges

The main goals of the CDCs are the following:

- integrate and test the most recent version of all the components of the DAQ fabric and the DAQ software, its interface with the ALICE offline software, and the Mass Storage System (MSS);
- measure the performance of the complete setup over long periods.

The performance measurements are executed with simulated data pushed through the DATE software framework. DATE performs a coordinated injection of the simulated data, and the event building. The data are then formatted with the DATE recorder based on the ROOT I/O package and AliRoot. A data catalogue is also created during the CDC. This catalogue is based on the Grid software (AliEn). The Mass Storage System (MSS) used is CASTOR. During these tests, the following items are measured:

- scalability of the DAQ system to control and configure hundreds of computing nodes;
- sustained aggregate throughput of the data transfer and the event-building inside the DAQ for several hours;
- sustained aggregate throughput data of the whole chain including recording to the permanent data storage system for seven consecutive days;
- global amount of data being recorded to the permanent data storage system.

The latest available hardware and software version of the following components have been tested:

- network technologies: trunking, backbone switching, Gigabit Ethernet and 10-Gigabit Ethernet;
- commodity hardware: hosts, network interface cards, tape units, and tape robots;
- Several DATE versions have been used to simulate the behaviour and the output of an ALICE-like experiment;
- ALICE fabric monitoring software (AFFAIR) [1] (page 245) to assess the behaviour of the components of the DAQ and the interface to the MSS;
- ALICE Offline software: objectification of raw data, handling of event objects, recording and interfacing to the MSS;
- CASTOR—deployed on CPU servers, disk servers and tape servers—for MSS functions;
- operating system (Linux) with its kernel, system, user libraries, drivers, file systems (local and networked), network daemons (standard and custom-designed).

Data traffic in use

The data used during the CDCs are simulated physics raw data for most of the ALICE detectors. These data are produced with the ALICE simulation program AliRoot [13] before the beginning of the Data Challenge, then split into several sub-events and injected by several computers into the data acquisition fabric.

For the purposes of the CDC, data are created by reading special data files into the *readout* process memory space during the start-of-run phase and then events are created using these data, as directed by a fixed configuration sequence. The data flow from the LDCs is created using the Configurable LDC Emulator (COLE) package. By configuring the sequence of events, their data, and their characteristics, it is possible to create several different patterns of data traffic: the ALICE data traffic, the ALICE events, and the ALICE simulated events.

- The ALICE data traffic is characterized by all LDCs sending a realistic amount of data, similar to that which they are supposed to send under ALICE running condition. The LDCs, their Network Interface Cards (NIC), and up-links handle realistic data blocks while the GDCs and their associated network resources are less loaded as they are supposed to be in real-life conditions. On account of the number of available LDCs (40 instead of 200 in the final DAQ system), one fifth of the ALICE nominal capacity could be simulated so far. The GDCs therefore handle events five times smaller than the real ALICE events.
- The ALICE events pattern creates at the output of the GDCs events whose size is equivalent to that which we expect at the same level under normal ALICE running condition. In this scenario, the LDCs create and handle events five times bigger than in real life while the GDCs write as much data as they are expected to do during ALICE heavy-ion data-taking periods.
- Finally, the ALICE simulated events scenario introduces a set of simulated events in the data stream. In the present implementation, several events are available for the same trigger classes and this varies the data flow coming from the LDCs quite considerably. During CDC V, this scheme was adopted for selected types of events (central and semi-central) and for some of the ALICE detectors (TPC, SDD, SSD, and SPD) for which simulated data are available.

For the three scenarios described above, a fixed sequence of events and trigger types that follows a distribution similar to that which is expected within the ALICE experiment during heavy-ion runs is created. This sequence implies the participation of selected detectors according to the trigger classes associated with each event and partial read-out is also applied for dielectron events.

1.3.2 Performances and results

Results of CDC III and CDC IV can be found in Refs. [14, 15]. Here we report in detail the objectives and the results of the last two CDCs, CDC V, held in 2003 and CDC VI that took place at the end of 2004 and beginning of 2005.

Figure 1.5 shows the planning of the ALICE Data Challenges as a function of the targeted data rates through the Data-Acquisition system (DAQ bandwidth) and recorded by the Mass Storage (Mass Storage bandwidth). The expected available tape bandwidth in the Tier 0 is also shown.

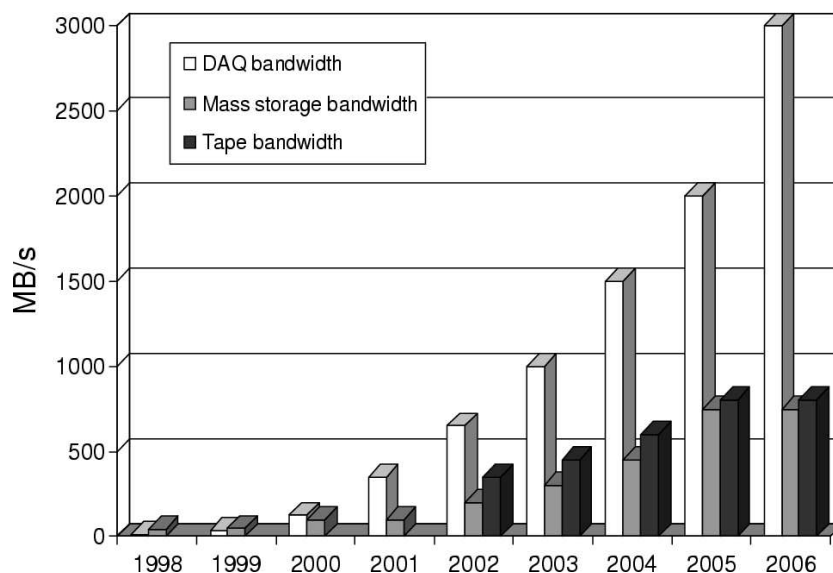


Figure 1.5: ALICE Data Challenge bandwidth planning.

As indicated in Fig. 1.5, the target is to increase the data rates progressively until something as close as possible to the final ALICE requirements—in agreement with the available hardware resources allocated to the exercise—is met. This will happen no later than one year before LHC startup.

As an example of the successfully reached milestones, Fig. 1.6 shows the sustained throughput achieved during CDC IV in 2002. The final results were a peak rate of 310 MB/s, a sustained average data rate produced by the DAQ System of 280 MB/s, and 180 TB moved onto the PDS over a period of seven consecutive days. The CDC IV was also a large-scale test combining 32 and 64 bits machines.

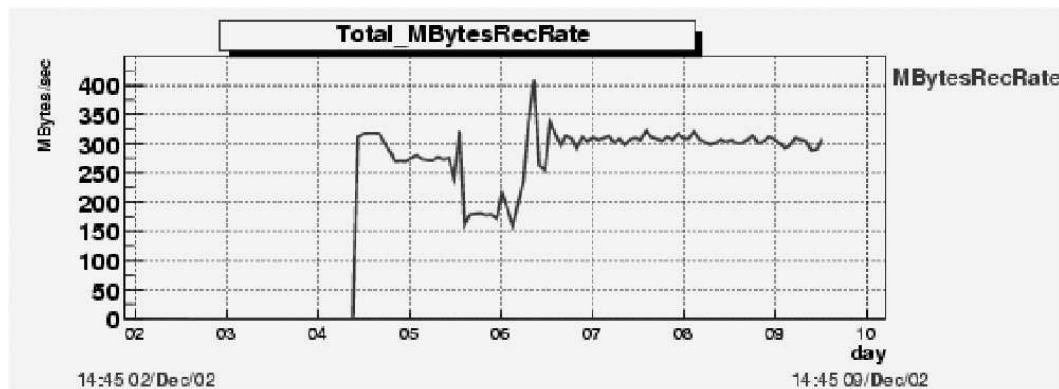


Figure 1.6: Sustained throughput milestone monitoring.

During CDC VI in 2005, higher sustained throughputs have been achieved with 2.5 GB/s of DAQ bandwidth (see Fig. 1.7) and an average of 450 MB/s to PDS with the RFIO format (see Fig. 1.8).

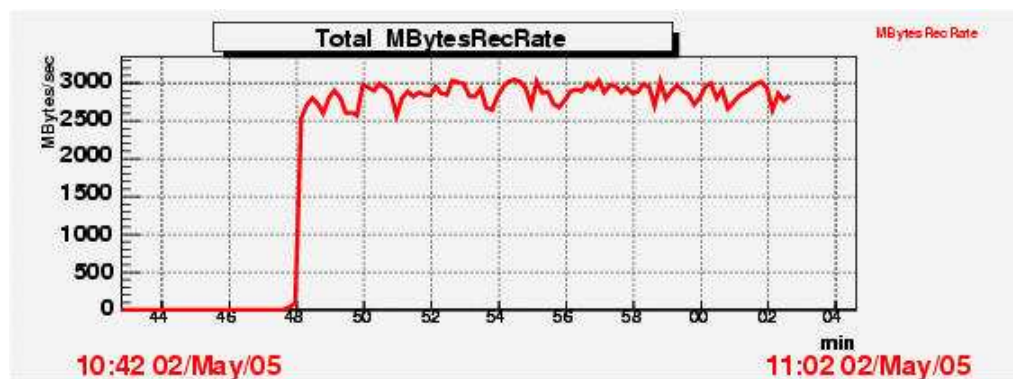


Figure 1.7: Sustained throughput of the DAQ system.

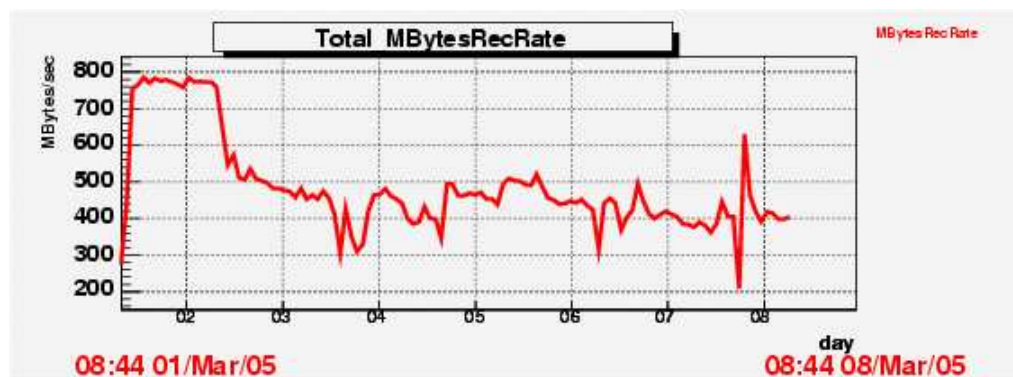


Figure 1.8: Sustained throughput to CASTOR.

Another important task during the CDCs is the online physics monitoring. The online physics monitoring is a software module developed within the AliRoot program framework. It runs fast physics event-reconstruction based completely on the available HLT software. The two main goals of the monitoring are to check the consistency of the raw data coming from the DAQ system and to test the performance of the HLT algorithms in an environment as realistic as possible. The monitoring code is run in two basic modes: online and quasi-online. In the online mode, the simulated raw data, after the event building, is processed and the HLT event summary data is written together with the raw data itself. The preliminary tests show very good time performance of the monitoring—ranging between 5 s and 10 s for processing and reconstruction of the TPC and the ITS raw data for central Pb–Pb events. The tracking efficiency depends on the HLT algorithms used and in general is about 5–10% lower than that of the standard ALICE offline reconstruction. The second operation mode of the online physics monitoring—the quasi-online one—makes use of the same HLT code, but in addition also runs some parts of the offline reconstruction chain. In this mode, the monitoring takes the files with raw data events already stored and registered on the Grid, processes them, and fills series of monitoring histograms. Owing to the flexibility of the developed framework, the user can be provided with fast feedback related to the quality of the raw data as well as to the physics performance of an individual sub-detector or the entire ALICE setup.

1.3.3 Future plans

The goals of the next Data Challenge (CDC VII) include the following points:

- realize the test in real conditions with the DAQ setup at the experimental area (LHC point 2) and the Tier 0 equipment in the computing centre;
- record the data in ROOT format;
- increase the number of pieces of equipment to be read;
- improve the profile of the data traffic including realistic trigger;
- add the contribution from hardware-driven data sources (such as the DDL data pattern generator);
- test new network technologies and topologies;
- further develop and optimize the online physics monitoring framework;
- test the online physics monitoring software for other ALICE sub-detectors;
- achieve higher performance.

References

Chapter 1

- [1] ALICE Collaboration, *Technical Design Report of Trigger, Data Acquisition, High-Level Trigger and Control System*, CERN/LHCC/2003-062.
- [2] The ALICE Trigger Project, ALICE Central Trigger Processor, User Requirement Document, Draft 1.0, 24 October 2001.
- [3] The ALICE Trigger Project, ALICE Local Trigger Unit, Preliminary Design Review Document, Revision 0.1, 1 September 2002.
- [4] B. G. Taylor, LHC machine timing distribution for the experiments, in *Proc. of the 6th Workshop on Electronics for LHC Experiments*, Cracow, Poland, 2000 (CERN 2000-010, Geneva, 2003).
- [5] J. Baud *et al.*, CASTOR status and evolution, in *Proc. Conf. on Computing in High-Energy Physics*, La Jolla, CA, USA, March 2003 (SLAC, Stanford).
- [6] ALICE DAQ, DATE V4 User's Guide, Internal Note ALICE-2002-036.
- [7] R. Divià *et al.*, Data Format over the ALICE DDL, Internal Note ALICE-INT-2002-10 V4.
- [8] ALICE DAQ, ALICE DDL Interface Control Document, Internal Note ALICE-INT-1996-43 V9.1.
- [9] ALICE DAQ, ALICE DDL Hardware Guide for the Front-end Designers, Internal Note ALICE-INT-1998-21 V1.3.
- [10] J. Christiansen *et al.*, TTCrx Reference Manual A Timing, Trigger and Control Receiver ASIC for LHC detectors, V 3.8, RD12 project, January 2003.
- [11] B. G. Taylor, TTC Machine Interface (TTCmi) User Manual, Rev 1.3, RD12 project.
- [12] P. Farthouat and P. Gällnö, TTC-VMEbus Interface TTCvi MkII, Rev 1.1, RD12 project, May 2000.
- [13] R. Brun *et al.*, *Computing in ALICE*, Nucl. Instr. Meth. **A502** (2003) 339-346.
- [14] W. Carena *et al.*, Online performance monitoring of the Third ALICE Data Challenge (ADC III), in *Proc. Conf. on Nuclear Electronics & Computing*, Varna, Bulgaria, September 2001.
- [15] T. Anticic *et al.*, Challenging the challenge: handling data in the Gigabit/s range, to be published in *Proc. Conf. on Computing in High-Energy Physics*, La Jolla, CA, USA, March 2003 (SLAC, Stanford).

