



UNIVERSITÀ
DI PAVIA

Dipartimento di Fisica “Alessandro Volta”

Corso di laurea magistrale in Scienze fisiche

Upgrade of the ATLAS Dataflow for the High-Luminosity LHC and Trigger-Level Analysis Workflows

Tesi per la laurea di:
Roberto Mastrofrancesco

Relatore:
Prof. Andrea Negri

Correlatore:
Dr. Matias Alejandro Bonaventura

Anno accademico 2025 – 2026

The organization is watching us. They
have secret agents everywhere.

Rintaro Okabe

Abstract

Italiano: La presente tesi ha come obiettivo principale lo sviluppo e la validazione del sistema dataflow dell'esperimento ATLAS in vista dell'High-Luminosity LHC (HL-LHC), nonché lo studio dell'integrazione delle tecniche di Trigger-Level Analysis (TLA) nel sistema aggiornato. A partire da una descrizione generale di LHC e dell'esperimento ATLAS, vengono presentati le caratteristiche principali e i requisiti del relativo sistema di trigger e acquisizione dati, sia nella configurazione di Run 3 sia alla luce degli aggiornamenti previsti per HL-LHC. Si illustra quindi l'attività svolta nell'ambito del gruppo dataflow di ATLAS: dallo sviluppo di un'interfaccia di compatibilità che consente di testare i nuovi sistemi di readout con il dataflow di Run 3, fino alla distribuzione di applicazioni di dataflow organizzate in ambienti di esecuzione isolati e autosufficienti (container), validate mediante un test su larga scala al rate di eventi previsto per Run 4. Infine, l'impatto delle metodologie TLA previste per HL-LHC sul sistema dataflow viene valutato attraverso test dedicati sull'infrastruttura di Run 3, mentre uno studio preliminare, basato su simulazioni Monte Carlo, di un bosone Z' leggero prodotto con radiazione di stato iniziale evidenzia la sensibilità che la TLA potrebbe garantire nel corso di Run 4.

English: The main goal of this thesis is the development and validation of the ATLAS dataflow system for the High-Luminosity LHC (HL-LHC), together with the study of Trigger-Level Analysis (TLA) workflows in the upgraded system. Starting from a general description of the LHC and of the ATLAS experiment, the main characteristics and requirements of its trigger and data acquisition system will be presented, both in its Run-3 version and in view of the planned HL-LHC upgrades. The work carried out within the ATLAS dataflow group will then be illustrated, from the development of a compatibility interface allowing upgraded readout systems to be tested with the Run-3 dataflow to the deployment of containerized dataflow applications, validated in a full-scale test at the Run-4 target event rate. Finally, the impact of the TLA workflows planned for the HL-LHC on the dataflow system will be assessed with dedicated tests on the Run-3 infrastructure, and a truth-level study of a light Z' boson produced with initial-state radiation will show the sensitivity that TLA could provide during Run 4.

Contents

Introduction	1
1 The Large Hadron Collider	3
1.1 The Accelerator Complex	3
1.1.1 Injection Chain	4
1.1.2 Machine Layout and Experiments	5
1.2 Luminosity and Operation	8
1.2.1 Luminosity and Event Rate	8
1.2.2 Pile-up	11
1.2.3 Delivered Luminosity	12
1.3 The High-Luminosity LHC	13
1.3.1 Motivation	13
1.3.2 Machine Upgrades	13
1.3.3 Target Parameters and Schedule	15
2 The ATLAS experiment	17
2.1 Detector Layout	18
2.1.1 Coordinate System	19
2.1.2 The Magnet System	20
2.2 The Inner Detector	21
2.2.1 Silicon detectors	21
2.2.2 Transition Radiation Tracker	22
2.2.3 Tracking Performance and the ITk	23
2.3 The Calorimeter System	24
2.3.1 Electromagnetic Calorimeter	25
2.3.2 Hadronic Calorimeter	25
2.3.3 Energy Resolution and Readout	26
2.4 The Muon Spectrometer	27
2.4.1 Precision Chambers	28

2.4.2	Trigger Chambers	29
2.4.3	New Small Wheels	29
2.5	Forward Detectors and Luminosity Measurement	30
2.5.1	Forward Detectors	30
2.5.2	Luminosity Measurement	32
2.6	Object Reconstruction	32
2.6.1	Jets	33
2.6.2	Other Physics Objects	35
2.6.3	Simulation	36
3	The ATLAS Trigger and Data Acquisition System	37
3.1	Trigger and Data Acquisition Principles	37
3.1.1	Dead Time and Buffering	38
3.1.2	Pipelined and Multi-Level Triggers	40
3.1.3	Event Building	42
3.2	Requirements	44
3.3	The Run 3 TDAQ System	45
3.3.1	The Level-1 Trigger	45
3.3.2	Readout and Dataflow	48
3.3.3	High-Level Trigger	51
3.4	The Run-4 TDAQ System	53
3.4.1	The Level-0 Trigger	53
3.4.2	Readout and Dataflow	54
3.4.3	Event Filter	55
4	The HL-LHC Dataflow Upgrade	57
4.1	The Dataflow Software	57
4.1.1	Software Architecture	58
4.1.2	Controls and Configuration	58
4.1.3	Testbeds	61
4.2	Integration of the Upgraded Run-4 Readout with the Run-3 Dataflow	62
4.2.1	Input Sources of the HLTSV	63
4.2.2	TTC Information in Run-4	64
4.2.3	The FELIX Input Source	65
4.2.4	Validation and Use	66
4.3	Deployment of the Event Builder on Kubernetes	67
4.3.1	Containers and Kubernetes	68
4.3.2	The Containerized Event Builder	71
4.3.3	Deployment Models	74
4.3.4	Deployment Tests	76
4.3.5	Health Monitoring of the Event Filter	79

5	Trigger-Level Analysis	82
5.1	Trigger-Level Analysis in ATLAS	82
5.1.1	Motivation	83
5.1.2	The TLA Workflow	84
5.1.3	Results	85
5.2	TLA Workflows in Run 4	86
5.2.1	Run-4 Scenarios	86
5.2.2	Impact on Dataflow	87
5.2.3	Test Setup	88
5.2.4	Results	90
5.3	The Z' Benchmark Model	91
5.3.1	Simplified Dark Matter Models	91
5.3.2	Production with Initial-State Radiation	95
5.3.3	Boosted Decay and Jet Substructure	97
5.3.4	Backgrounds	100
5.4	Truth-Level Sensitivity Study	100
5.4.1	Samples and Selection	101
5.4.2	Signal Kinematics	103
5.4.3	Signal Reconstruction	104
5.4.4	Background	105
5.4.5	Sensitivity Estimate	108
	Conclusion	113
	Bibliography	115

Introduction

The *Large Hadron Collider* (LHC) at CERN is the largest and most energetic particle accelerator in the world. It has been delivering both proton-proton and heavy ion collisions at unprecedented energies and luminosities since 2009. Currently, the third long shutdown period (LS3) is ongoing, during which the LHC and its experiments are being upgraded to the *High Luminosity LHC* (HL-LHC). Starting from 2030, data-taking will resume with the fourth run period (Run 4) and the LHC will reach an instantaneous luminosity five times larger than the original design value.

The main goal driving research at the LHC has been and remains the study of extremely rare physics processes, with the final intent of verifying theoretical models. To achieve this, LHC experiments have to operate a real-time selection of about one event among $\mathcal{O}(10^4)$ total events, ensuring minimal latency and downtime. Trigger and Data Acquisition (TDAQ) systems performance is therefore crucial to exploit the full physics potential of the LHC.

This challenge becomes even more demanding when considering the HL-LHC conditions. To cope with the higher rates and larger events, the TDAQ systems of the LHC experiments will have to be extensively upgraded. In the ATLAS experiment, subject of this thesis, the upgraded dataflow system has to be developed and tested at full scale before the start of Run 4, also supporting new workflows, such as *Trigger-Level Analysis* (TLA) ones. TLA has proved to be a valuable technique, allowing to record at very high rates the objects already reconstructed by the trigger for events that would otherwise be lost.

The first chapter (chapter 1) provides a general overview of the LHC complex, introducing the main physics goals and important concepts such as luminosity and pile-up. It also illustrates the HL-LHC upgrade program, from its motivations to the expected performance. Next, chapter 2 describes the ATLAS experiment in detail, focusing on the detector layout and explaining the various subdetector

characteristics. Additionally, jet-specific reconstruction and calibration techniques are discussed, as they are relevant to the work presented later in the thesis.

In chapter 3 the ATLAS TDAQ system is analyzed in detail, starting from trigger and data acquisition principles and common techniques, then moving to their implementation in the ATLAS experiment. The chapter concludes with a description of the HL-LHC planned upgrades, highlighting major changes and new features.

Later, chapter 4 presents the HL-LHC design of the ATLAS dataflow system. More specifically, the chapter focuses on the work I carried out at CERN during my research internship, starting from the development of a testing interface to ease-out upgraded subdetector integration and validation, to the actual contribution I gave to the planned containerization of some dataflow components. The chapter concludes with a discussion of the results obtained during a full-scale testing campaign where the upgraded dataflow system was validated and its performance measured.

Finally, chapter 5 describes the work I carried out together with the ATLAS TLA group, from a feasibility test of the integration of TLA workflows in the upgraded dataflow system, up to the analysis of a physics case that could benefit from TLA in Run 4: the search for a light Z' boson decaying into quarks and produced with initial-state radiation.

Chapter 1

The Large Hadron Collider

The *Large Hadron Collider* (LHC) is currently the world's largest and most powerful particle accelerator, located at CERN, the *European Organization for Nuclear Research*, near Geneva, Switzerland. Placed in the older *Large Electron-Positron Collider* (LEP) tunnel, the LHC mainly consists of a circular accelerator 27 kilometers long, spanning across the France Switzerland border. It was designed to collide protons at $\sqrt{s} = 14$ TeV, allowing for precision measurement of *Standard Model* (SM) physics, as well as helping understand dark matter and beyond the SM physics.

On July 4, 2012, the ATLAS and CMS experiments at the LHC announced the discovery of a new particle with a mass of ≈ 125 GeV: the Higgs boson, a key component for the SM to work. Its discovery brought the 2013 Nobel Prize in Physics to François Englert and Peter Higgs “*for the theoretical discovery of a mechanism that contributes to our understanding of the origin of mass of subatomic particles, and which recently was confirmed through the discovery of the predicted fundamental particle, by the ATLAS and CMS experiments at CERN's Large Hadron Collider.*” [1]

1.1 The Accelerator Complex

At CERN, the LHC is part of a larger accelerator complex that includes several other accelerators and detectors. Experiments and colliders have been upgraded, swapped or replaced over the years to reach the designated physics goals and the current layout can be found in figure 1.1.

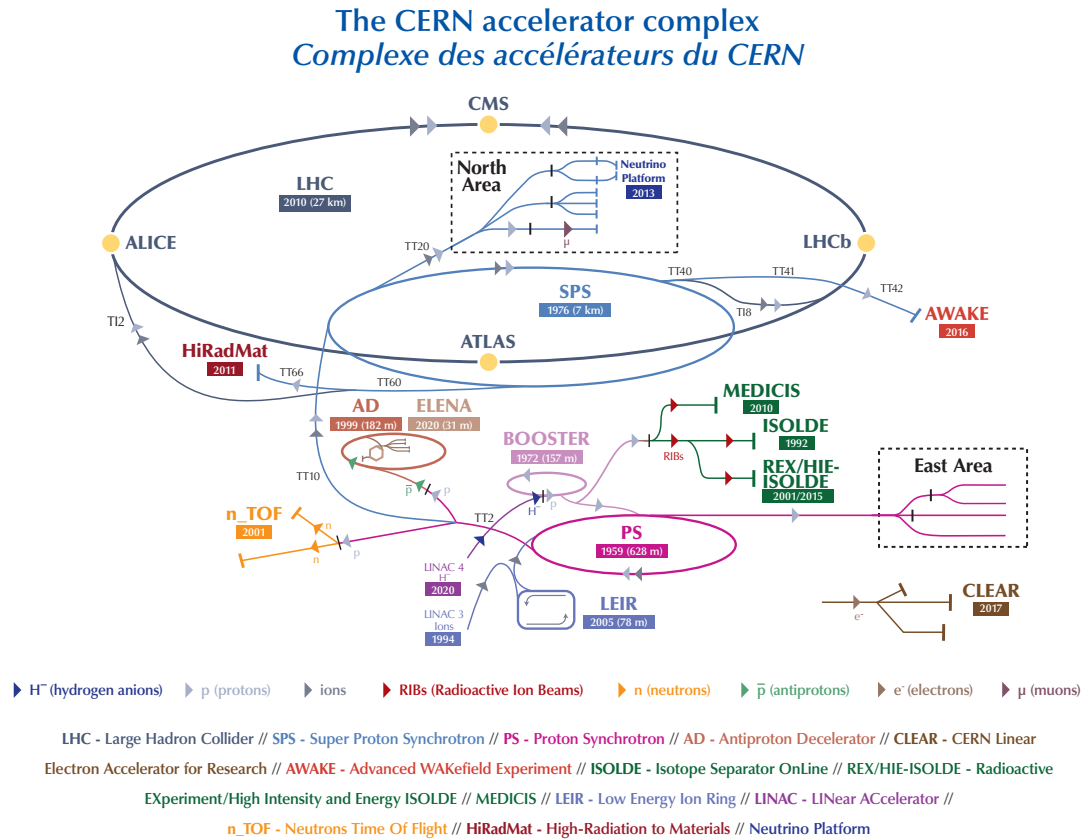


Figure 1.1: The CERN accelerator complex [2].

1.1.1 Injection Chain

To properly reach the designated energy levels, the LHC requires a series of pre-accelerators to boost the particles' energy before they are injected into the main ring. The chain begins with a plasma source producing negative hydrogen ions (H^-), which a first linear accelerator, *LINAC 4*, accelerates to 160 MeV. At injection into the *Proton Synchrotron Booster* (PSB), the ions traverse a thin carbon foil which strips both electrons, leaving only protons; the PSB then accelerates them to 2 GeV before they enter the *Proton Synchrotron* (PS), which brings them to 26 GeV. Finally, the *Super Proton Synchrotron* (SPS) raises the energy to 450 GeV for injection into the LHC ring.

Filling the LHC with protons is a complex process that requires multiple injections at each step of the chain. The PS groups the protons into batches of 72 bunches spaced by 25 ns, up to four of which are accumulated in the SPS; twelve SPS

injections are then needed to fill each LHC ring. Some of the 3564 available bunch slots are deliberately left empty to allow for the rise time of the injection and beam dump kicker magnets, the largest of these gaps being the 3 μs *abort gap* [3, 4].

Since a typical SPS cycle lasts 21.6 s, filling one LHC ring takes just under five minutes in ideal conditions, while the complete injection process for both beams normally requires 20 to 30 minutes [5]. The bunches' injection scheme described above can be found in figure 1.2.

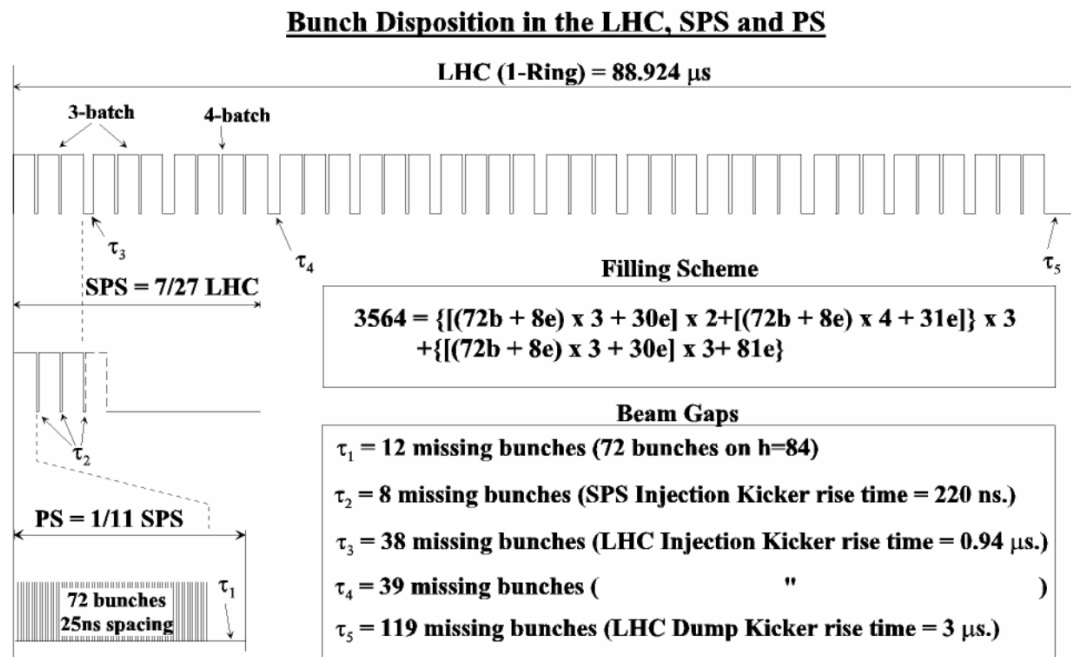


Figure 1.2: Bunches in the LHC, SPS and PS machines [5].

1.1.2 Machine Layout and Experiments

The LHC accelerates protons in two counter-rotating beams, so twin-bore dipoles with opposite field directions in the two apertures are used to bend them in a circular path; RF cavities are used to accelerate them, while quadrupoles and sextupoles focus the beams, all installed in the former LEP tunnel.

The tunnel itself consists of eight straight sections, 528 m each, and eight arcs. This shape was chosen originally to allow long accelerating sections for the LEP, to compensate the synchrotron radiation losses, negligible in the case of protons. The power emitted due to the acceleration produced by the bending of the beams

can be expressed as:

$$P = \frac{e^2 c}{6\pi\epsilon_0 R^2} \gamma^4 = \frac{e^2 c}{6\pi\epsilon_0 R^2} \left(\frac{E}{mc^2} \right)^4 \quad (1.1)$$

where P is the power radiated, c the speed of light, ϵ_0 the vacuum permittivity, γ the Lorentz factor, R the bending radius and e , E and m the particle charge, energy and mass, respectively. The energy lost per revolution then follows as

$$\Delta E = PT = P \frac{2\pi R}{v} \approx \frac{e^2}{3\epsilon_0 R} \left(\frac{E}{mc^2} \right)^4 \quad (1.2)$$

Taking $R = 2804$ m, the bending radius inside the dipoles, a proton at 7 TeV loses about 6.7 keV per turn, while an electron at the same energy would lose approximately 7.6×10^7 GeV, four orders of magnitude more than its own energy.

The current LHC layout is shown in figure 1.3: the four interaction points (IPs) are located at the center of four of the straight sections, where the main experiments are placed: ATLAS, CMS, ALICE and LHCb. The beams are injected into the two rings at the ALICE and LHCb points, while the remaining four straight sections are used for beam collimation, radiofrequency acceleration and beam diagnostics, and beam extraction towards the dump.

ATLAS and CMS are general-purpose detectors, placed at the two highest luminosity interaction points and designed to cover the widest possible range of physics, from precision Standard Model measurements to searches for new phenomena. ALICE is dedicated to the study of heavy-ion collisions and of the quark-gluon plasma, while LHCb is optimized for flavor physics and instruments only the forward region, where b -hadron production is enhanced (because the produced b quarks are boosted, thus the resulting b -hadrons have a bigger decay length $\beta\gamma c\tau$). The ATLAS detector, on which the work presented in this thesis is based, is described in detail in chapter 2.

The modest radiation losses also relax the requirements on the accelerating system: where LEP needed an accelerating voltage of the order of a gigavolt to replace the energy radiated by its beams, the LHC requires only about 16 MV per beam. The acceleration is provided by eight single-cell superconducting cavities per beam, operating at 400 MHz and housed in cryomodules in the straight section at Point 4.

The cavities ramp the beams from 450 GeV up to the collision energy and define RF buckets spaced by 2.5 ns, one every ten of which is filled, giving the 25 ns bunch spacing described in section 1.1.1.

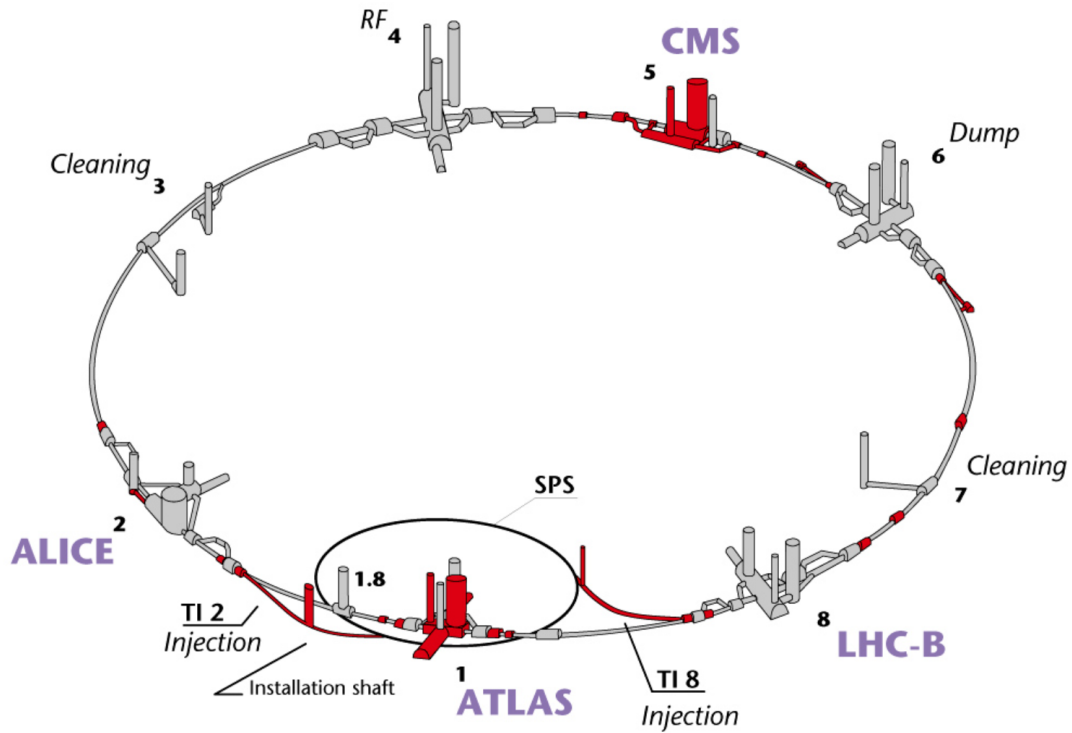


Figure 1.3: The LHC tunnel layout [6]. In red the new buildings added for the LHC, while in gray the pre-existing LEP infrastructure.

As briefly mentioned above, to perform the bending the LHC uses a total of 1232 twin-bore superconducting dipole magnets, sharing the same cryostat and cooled to 1.9 K with superfluid helium. Additionally 392 quadrupoles are placed along the ring to focus the beams, while sextupoles, octupoles and decapoles are used to correct errors in the particle trajectories.

The usage of helium to reach the superconducting regime is necessary because no ordinary magnets can produce the required magnetic field of 8.3 T. This value can be trivially derived using the following relation:

$$B = \frac{p}{eR} \quad (1.3)$$

where B is the magnetic field, $p = \sqrt{E^2 - m^2c^4}$ the particle momentum, e the particle charge and R the bending radius. For a proton at 7 TeV in a 2804 m bending radius, the required field turns out to be ≈ 8.3 T. The alloy used for the superconducting cables to allow for such high currents is *Niobium-Titanium* (NbTi), with a critical temperature of 9.2 K; the dipole structure can be found in figure 1.4.

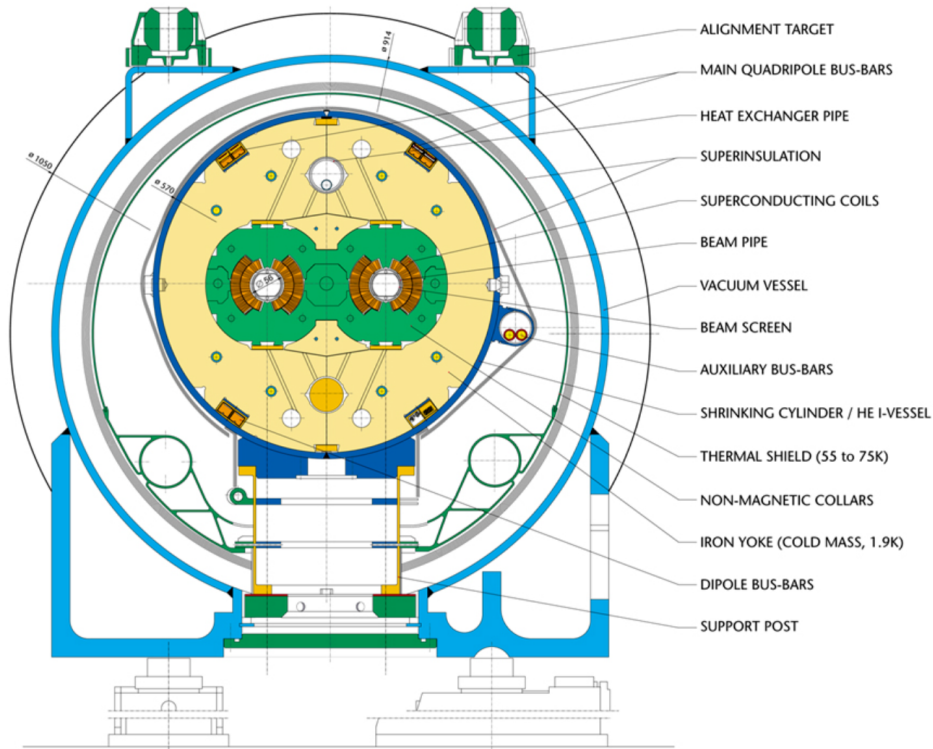


Figure 1.4: The LHC dipole magnet cross-section [7].

1.2 Luminosity and Operation

The performance of a collider is quantified by its luminosity, which together with the cross-section of a given process determines the rate at which it is produced. The same quantity also fixes the number of simultaneous interactions per bunch crossing, and therefore the complexity of the events that the detectors and the readout system have to handle. The following sections introduce these quantities and report the values achieved by the LHC so far.

1.2.1 Luminosity and Event Rate

The rate of events R generated in a collider for a given process is given by:

$$R = \sigma \mathcal{L} \quad (1.4)$$

where σ is the cross-section of the process and $\mathcal{L}$ the instantaneous luminosity of the accelerator. The latter normally depends on beam parameters and can be

defined as:

$$\mathcal{L} = \frac{N_1 N_2 n_b f_{rev} \gamma}{4\pi \epsilon_n \beta^*} F \quad (1.5)$$

where:

- N_1 and N_2 are the number of particles per bunch in the two beams;
- n_b is the number of bunches per beam;
- f_{rev} is the revolution frequency;
- γ is the Lorentz factor;
- ϵ_n is the normalized beam emittance, giving a measure of how much the beam spreads in space;
- β^* is the beta function, giving the envelope for the particle's motion at the collision point;
- F is the luminosity reduction factor, accounting for the crossing angle of the beams at the interaction point.

While f_{rev} and γ are fixed by the collider radius and particle energy, the other parameters can be tuned to maximize the luminosity.

Integrating the instantaneous luminosity over a data taking period gives the *integrated luminosity*:

$$\mathcal{L}_{\text{int}} = \int \mathcal{L} \, dt \quad (1.6)$$

which quantifies the size of the collected dataset and is commonly expressed in inverse femtobarns. The expected number of events of a given process then follows directly as $N = \sigma \mathcal{L}_{\text{int}}$, making $\mathcal{L}_{\text{int}}$ the figure of merit for any measurement limited by statistics.

Since the LHC physics program aims at studying extremely rare processes, as depicted in figure 1.5, showing the cross-section of various SM processes as a function of the center-of-mass energy, to obtain a statistically relevant rate the luminosity must be maximized (in accordance with (1.4)).

For this reason, the LHC was designed to reach a peak luminosity of $10^{34} \text{ cm}^{-2} \text{ s}^{-1}$, a value which consistently increased throughout the various machine runs, as shown in table 1.1. At the design center of mass energy, the total proton-proton

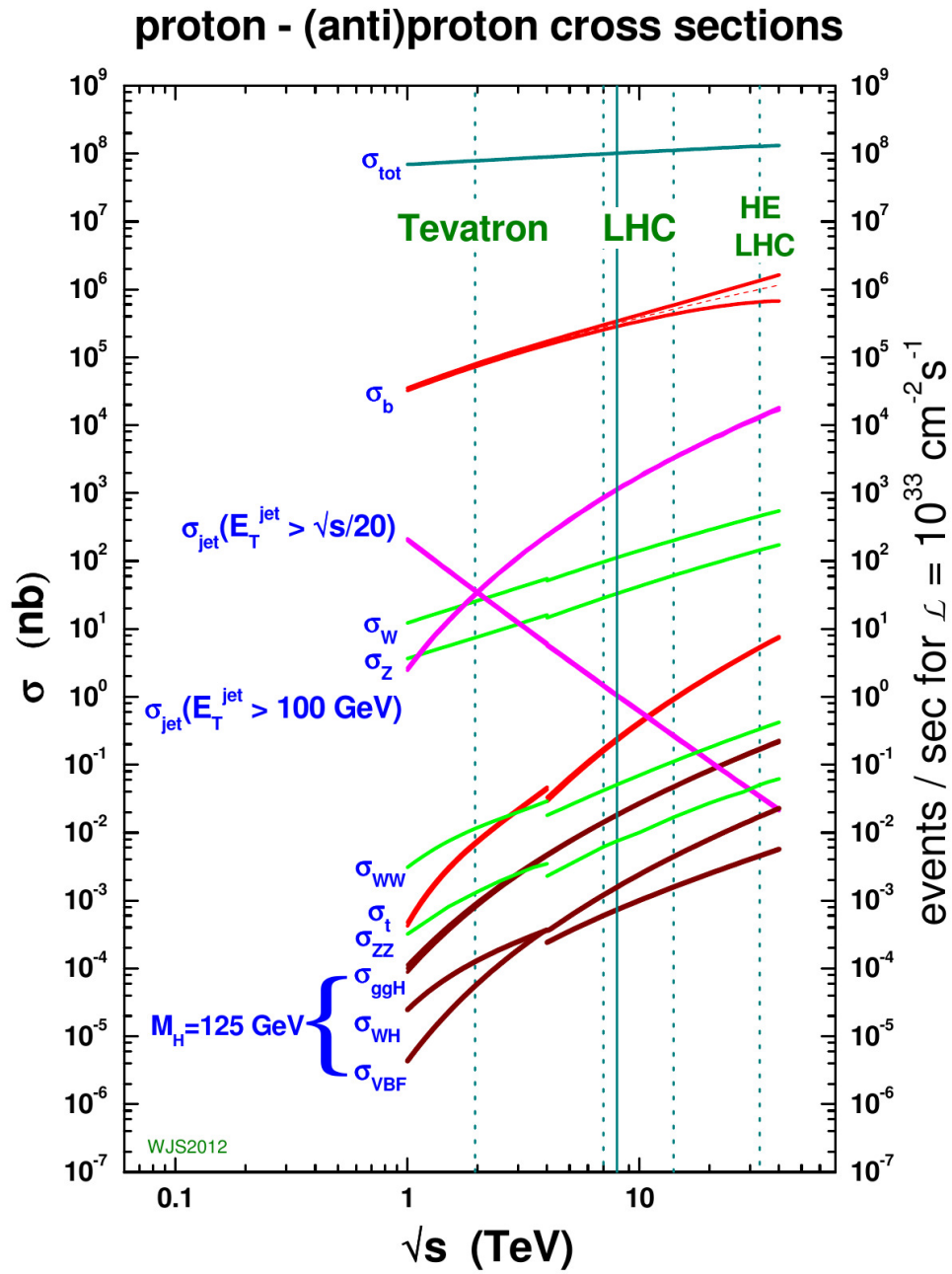


Figure 1.5: Cross-section of various SM processes as a function of the center-of-mass energy [8].

cross-section is $\sigma_{pp} \approx 100 \text{ mb}$, yielding $R \approx 10^9 \text{ Hz}$ events, most of which classifiable as *not of interest*. On the other hand, the Higgs boson production cross-section, being nine orders of magnitude smaller, results in $R_H \approx 0.25 \text{ Hz}$, highlighting the need for high luminosity to gather sufficient data to study rare processes.

Table 1.1: Summary of LHC parameters up to Run 4.

Time Period	$\sqrt{s}$	Peak luminosity	Bunch spacing	$\langle\mu\rangle$
Run-1 (2010–2012)	7–8 TeV	$0.8 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$	50 ns	40
Run-2 (2015–2018)	13 TeV	$1.7 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$	25 ns	25–50
Run-3 (2022–2026)	13.6 TeV	$2.0 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$	25 ns	< 60
Run-4 – HL-LHC (2030–2033)	14 TeV	$5.0 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$	25 ns	< 200

1.2.2 Pile-up

On each collision, actually multiple interactions occur, since each bunch contains billions of protons. The number of interactions per bunch crossing is called *pile-up*, and it's defined as:

$$\langle\mu\rangle = \sigma_{inel} \mathcal{L} \tau_b \quad (1.7)$$

where σ_{inel} is the total inelastic proton-proton cross-section, $\mathcal{L}$ the instantaneous luminosity and $\tau_b = (f_{rev} n_b)^{-1}$ the bunch spacing.

More precisely, (1.7) gives the *in-time pile-up*, i.e. the number of interactions occurring in the same bunch crossing. Another type of pile-up is the *out-of-time pile-up*, which refers to the interactions occurring in bunch crossings adjacent to the one being read out. Due to detector latencies, which can be bigger than the bunch spacing (25 ns), the readout system can integrate signals from multiple bunch crossings as if they were from the same one, thus increasing the number of interactions to be reconstructed. The total pile-up is then given by the sum of the in-time and out-of-time contributions. Table 1.1 reports the average pile-up values for the various LHC runs, which are expected to increase significantly in the HL-LHC era (section 1.3).

1.2.3 Delivered Luminosity

Not all produced events end up in the dataset used for physics analyses: the *delivered* luminosity refers to the events provided by the LHC while the beams are in collision; of this, only the fraction collected while the data acquisition system is active and correctly configured is saved, giving the *recorded* luminosity. A further selection is then applied offline, requiring all detector subsystems to have been fully operational, defining the luminosity that is *good for physics*.

The ratio between recorded and delivered luminosity is a direct measure of the data taking efficiency of the experiment, and therefore of the performance of the trigger and data acquisition system. Figure 1.6 shows the luminosity delivered to and recorded by ATLAS over the LHC runs.

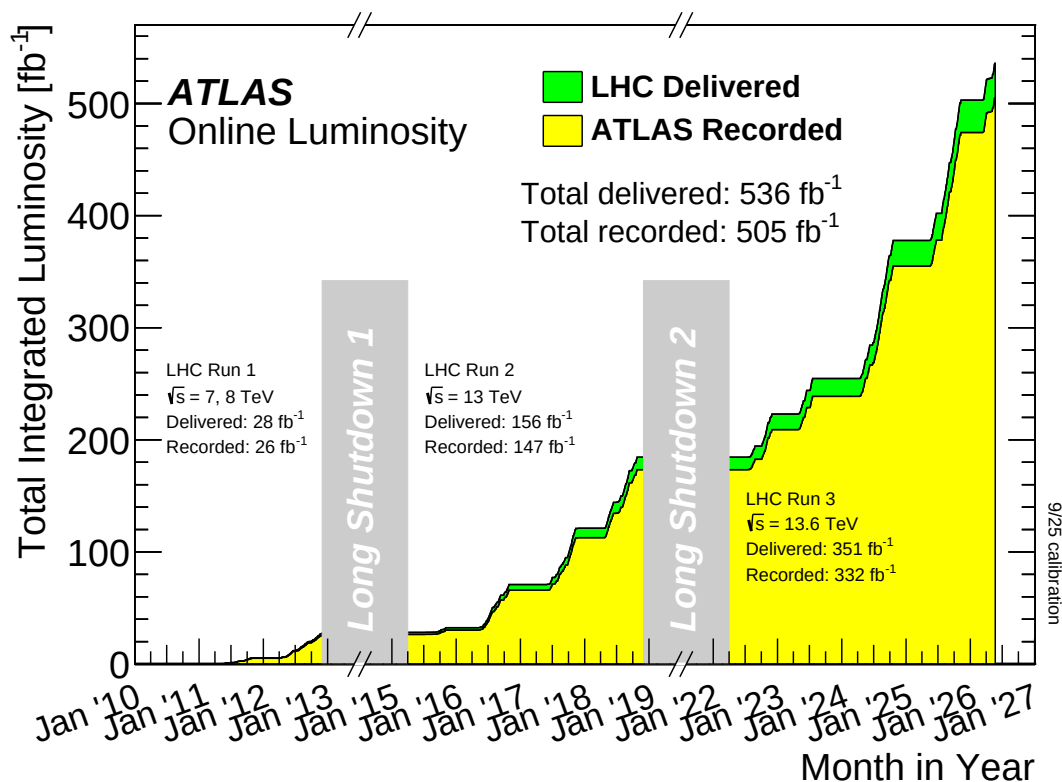


Figure 1.6: Luminosity delivered to and recorded by ATLAS over the LHC runs [9].

1.3 The High-Luminosity LHC

With the end of Run 3 the LHC reached a total delivered luminosity of about 450 fb^{-1} . Many of the processes of interest have cross-sections small enough that this dataset is not sufficient to study them, and since the collision energy cannot be increased significantly, the only way to collect more events is to increase the luminosity. The *High-Luminosity LHC* (HL-LHC) is the upgrade program planned for this purpose: it aims at an instantaneous luminosity five times larger than the original design value and at a tenfold increase of the integrated luminosity [10]. The following sections cover the motivation for the upgrade, the modifications to the machine, and the parameters and schedule of the new configuration.

1.3.1 Motivation

From (1.6) the number of events of a given process grows linearly with the integrated luminosity, so the statistical uncertainty on a measurement improves as $1/\sqrt{\mathcal{L}_{\text{int}}}$. This means that the tenfold increase planned for the HL-LHC corresponds to roughly a factor of three on the statistical precision. The improvement is largest for the measurements that are currently limited by the available statistics, while processes with cross-sections too small to be observed so far will become accessible for the first time.

An example of this improvements would be the measurement of the Higgs boson self-coupling through di-Higgs production, for which a precision of the order of 50 % per experiment is expected, improving to better than 30 % when ATLAS and CMS are combined. The larger dataset will also give access to rare Higgs decay channels, reduce the uncertainties on the Higgs couplings to the level of a few percent, and extend the mass reach of searches for new phenomena [11].

The projected evolution of the peak and integrated luminosity is shown in figure 1.7: even with the upgraded machine, the target of 3000 fb^{-1} is only reached towards the end of the program.

1.3.2 Machine Upgrades

Running at an average luminosity five times larger than the original LHC design value ($10^{34} \text{ cm}^{-2} \text{ s}^{-1}$, section 1.2.1) without levelling would require starting each fill from a peak about 15 times larger, which would produce a pile-up beyond what the detectors can reconstruct, and would consume the beam intensity within a few hours. The HL-LHC will therefore rely on *luminosity levelling*: the beams are kept partially separated at the interaction point and their overlap is progressively

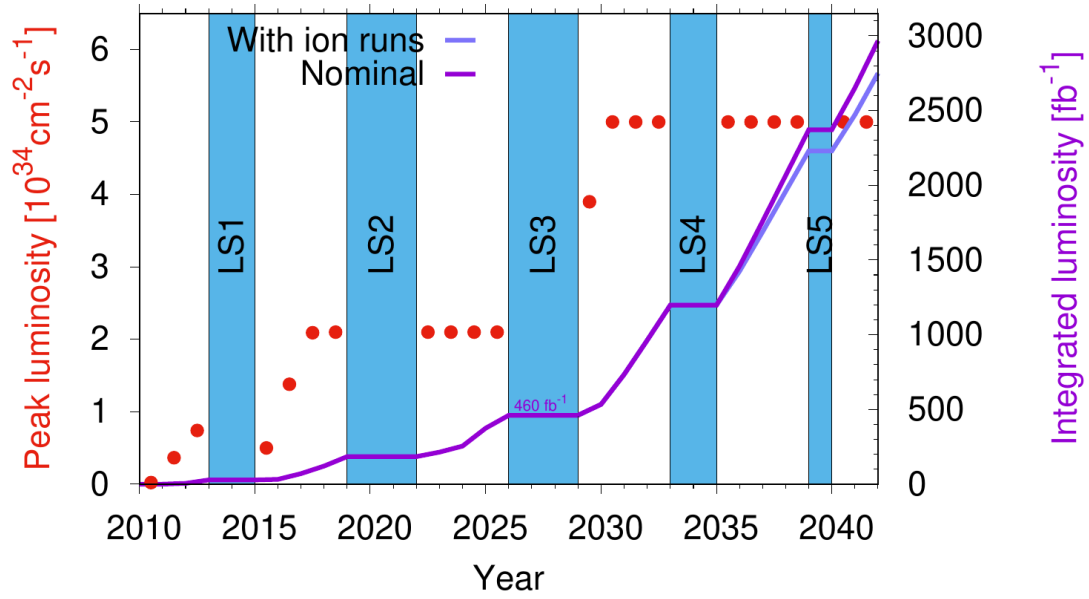


Figure 1.7: Projected evolution of the peak and integrated luminosity of the HL-LHC [10].

increased as the intensity decays, keeping the luminosity approximately constant at $5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ for most of the fill [10]. The effect of this technique is shown in figure 1.8.

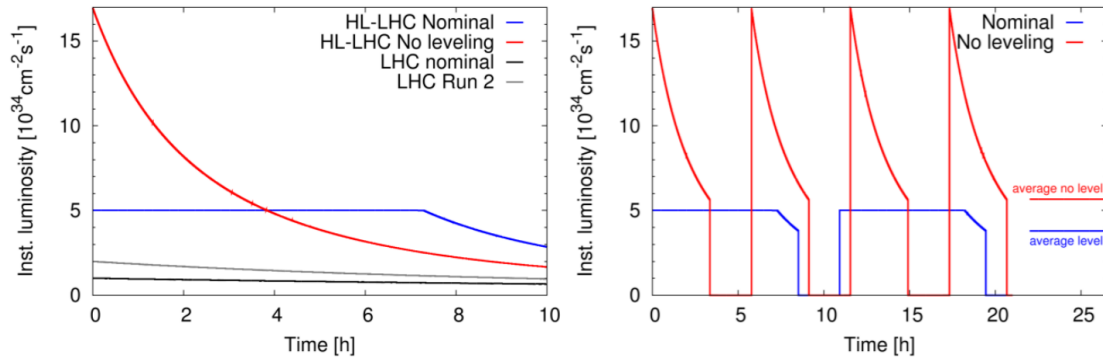


Figure 1.8: On the left, the luminosity profile for a single long fill for LHC nominal design and LHC Run 2 operation, as well for HL-LHC with and without levelling. On the right, the luminosity profile with optimized run time, without and with levelling, and average luminosity indicated in both cases (solid lines). From figure 1–2 in [10].

Reaching this regime requires a substantial modification of about 1.2 km of the machine around the two high-luminosity interaction points. The focusing quadrupoles

of the inner triplets will be replaced with Nb₃Sn magnets, capable of operating at fields beyond the limit of the NbTi technology described in section 1.1.2. Superconducting *crab cavities* will rotate the bunches before the collision so that they overlap head-on, recovering the luminosity otherwise lost to the crossing angle through the factor F of (1.5). On top of these upgrades a new optics scheme, an upgraded collimation system and superconducting links carrying the current to the magnets from outside the tunnel are also planned [10].

1.3.3 Target Parameters and Schedule

The resulting machine parameters are summarized in the previous table 1.1, together with those of Run 1–3. In the nominal scenario the levelled luminosity will reach $5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$, with an average pile-up of about 140, while the ultimate scenario foresees $7.5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ and a pile-up of up to 200 (this scenario refers to the maximum performance reachable if beam quality and hardware exceed expectations, with a total delivered luminosity of 4000 fb^{-1}) [10].

The installation of the new equipment will take place during the third Long Shutdown (LS3), which began in July 2026 and will last approximately four years; the first beams of Run 4 are expected in 2030 [12]. The overall timeline of the program, including the energy and luminosity milestones of each run, is shown in figure 1.9.



Figure 1.9: Timeline of LHC upgrades and modifications up to HL-LHC [12].

Chapter 2

The ATLAS experiment

The ATLAS (A Toroidal LHC ApparatuS) experiment [13] is the largest general-purpose particle detector at the Large Hadron Collider (LHC) at CERN, being 45 meters long, 25 meters in diameter and weighing more than 7000 tons. It was designed to gather data for a wide range of physics analyses, from precise measurements of Standard Model processes to searches for exotic new particles. To achieve this, it's important for the detector to have a high acceptance and efficiency for a wide range of particles, containing and detecting the highest number of particles produced during the collisions.

At the LHC, a bunch crossing occurs every 25 ns, each producing tens of simultaneous interactions, as described in section 1.2; to achieve a meaningful detection efficiency, the detector elements have to be finely segmented to distinguish the particles coming from different vertices, and their readout fast enough not to integrate the signals of consecutive crossings. Furthermore, from figure 1.5 it's clear that jet production largely dominates over the rare “interesting” processes ATLAS aims to study, thus requiring the detector to correctly reconstruct electrons, photons, muons, tau leptons, jets and missing transverse momentum, necessary to correctly identify the experimental signatures of the processes of interest. On top of these requirements, sensor and front-end electronics have to be built to withstand the radiation damage caused by the high particle flux. Finally, reading out the whole detector at 40 MHz would produce an amount of data impossible to store, so only a small fraction of the collisions is selected by the trigger and data acquisition system, described in chapter 3.

2.1 Detector Layout

The ideal shape for a detector with the requirements above is the cylindrical geometry, with the beam pipe running along the axis. Subdetectors are then arranged following this rationale, in concentric layers around the interaction point, each one measuring a different property of the particles crossing it. Both momenta and energy are measured, along with the particle trajectories, which are reconstructed from the hits in the tracking detectors. For neutrinos, which are weakly interacting and therefore invisible to the detector, the missing transverse momentum is inferred from the conservation of total transverse momentum in a single collision.

A detailed cutaway of the ATLAS detector is shown in figure 2.1. The main subsystems, which will be described in dedicated sections further on, are the inner detector, the electromagnetic and hadronic calorimeters and the muon spectrometer. ATLAS performance goals are summarized in table 2.1 (in terms of the variables defined in the following section 2.1.1).

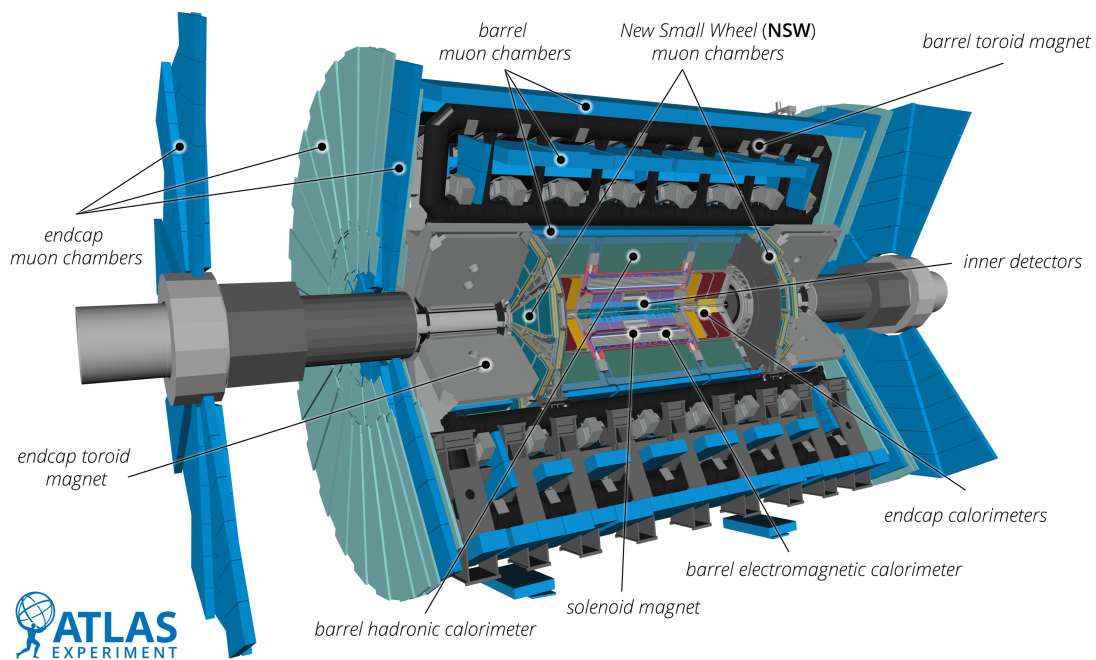


Figure 2.1: Cutaway of the ATLAS detector [14].

Table 2.1: General performance goals of the ATLAS detector, with E and p_T expressed in GeV. For high- p_T muons the performance of the muon spectrometer does not depend on the inner detector [13].

Detector component	Required resolution	η coverage	
		Measurement	Trigger
Tracking	$\sigma_{p_T}/p_T = 0.05\% p_T \oplus 1\%$	± 2.5	
EM calorimetry	$\sigma_E/E = 10\%/\sqrt{E} \oplus 0.7\%$	± 3.2	± 2.5
Jet calorimetry			
barrel & end-cap	$\sigma_E/E = 50\%/\sqrt{E} \oplus 3\%$	± 3.2	± 3.2
forward	$\sigma_E/E = 100\%/\sqrt{E} \oplus 10\%$	$\pm [3.1, 4.9]$	$\pm [3.1, 4.9]$
Muon spectrometer	$\sigma_{p_T}/p_T = 10\%$ at $p_T = 1$ TeV	± 2.7	± 2.4

2.1.1 Coordinate System

The origin of the ATLAS coordinate system is located at the nominal interaction point with the beams travelling along the z -axis. The x -axis points towards the center of the LHC ring, while the y -axis points upwards. The azimuthal angle ϕ is measured in the x - y plane around the beam axis, while the polar angle θ is measured from the z -axis. The distance from the z -axis is defined as the radial coordinate R . For a particle of momentum $\vec{p}$:

$$\phi = \arctan\left(\frac{p_y}{p_x}\right) \quad \theta = \arctan\left(\frac{p_T}{p_z}\right)$$

where $p_T = \sqrt{p_x^2 + p_y^2}$ is the *transverse momentum*; the *transverse energy* can be defined analogously: $E_T = E \sin \theta$.

In the ultrarelativistic approximation, where the particle mass is negligible with respect to its momentum, the rapidity y :

$$y = \frac{1}{2} \ln\left(\frac{E + p_z}{E - p_z}\right) \approx \frac{1}{2} \ln\left(\frac{|\vec{p}| + p_z}{|\vec{p}| - p_z}\right) = -\ln\left(\tan \frac{\theta}{2}\right) \equiv \eta$$

where η is the *pseudorapidity*, often used in place of the rapidity, as it can be calculated from the polar angle alone. The pseudorapidity is a convenient variable to describe the angular distribution of particles, as it is invariant under Lorentz boosts along the beam axis.

Often the distance itself is expressed in terms of angular separation in the η - ϕ plane, defined as:

$$\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$$

2.1.2 The Magnet System

The field needed to measure the momentum of charged particles is produced by four superconducting magnets:

- **one solenoid**, placed around the inner detector, provides an axial field of 2 T; its main purpose is to bend the trajectories of the charged particles inside the inner detector;
- **three toroids**, one in the barrel and one in each end-cap, placed outside the calorimeters; they produce a field of about 0.5 T in the barrel and 1 T in the end-cap region, giving a large bending power over a big volume while adding very little material, so that multiple scattering is limited and the momentum measured by the muon spectrometer is independent of the one measured by the inner detector.

The whole system is 22 m in diameter and 26 m in length, stores an energy of 1.6 GJ and produces a field over a volume of about 12 000 m³; its main parameters are summarized in table 2.2. All four magnets are operated at 4.5 K, less demanding than the 1.9 K needed by the LHC dipoles (section 1.1.2), since the fields involved are considerably lower. A schematic of the ATLAS magnet system is shown in figure 2.2.

Table 2.2: Main parameters of the ATLAS magnet system. [13]

	Solenoid	Barrel toroid	End-cap toroids
Inner diameter	2.46 m	9.4 m	1.65 m
Outer diameter	2.56 m	20.1 m	10.7 m
Axial length	5.8 m	25.3 m	5.0 m
Number of coils	1	8	2 × 8
Nominal current	7.73 kA	20.5 kA	20.5 kA
Stored energy	0.04 GJ	1.08 GJ	2 × 0.25 GJ
Peak field in the windings	2.6 T	3.9 T	4.1 T
Field range in the bore	0.9 T to 2.0 T	0.2 T to 2.5 T	0.2 T to 3.5 T

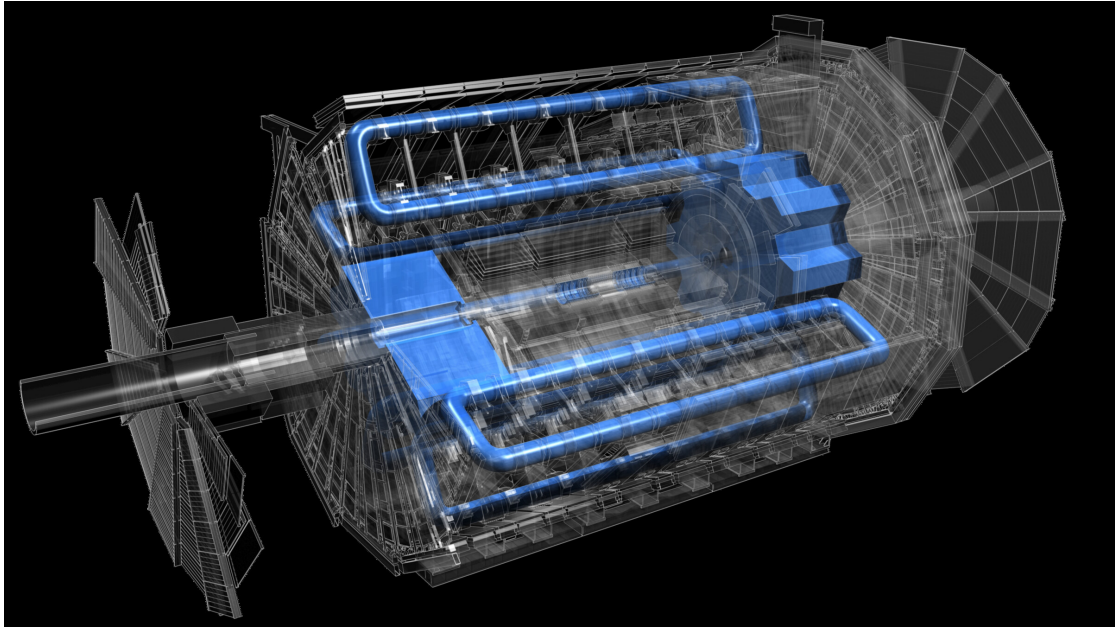


Figure 2.2: Schematic of the ATLAS magnet system [15].

2.2 The Inner Detector

The *Inner Detector* (ID) is the innermost part of ATLAS, the closest subdetector to the beam pipe. Using both silicon and gaseous detectors, its goal is to measure the trajectories of particles with precision $\mathcal{O}(10\ \mu\text{m})$ and to identify primary and secondary vertices. Moreover, as mentioned in section 2.1.2, the ID is immersed in a 2 T axial magnetic field which bends the particles. The total coverage provided is $|\eta| < 2.5$. A schematic of the inner detector is shown in figure 2.3.

2.2.1 Silicon detectors

Two types of silicon detectors are used in the inner detector:

- **pixel tracker**, the innermost element, designed to provide the highest granularity and precision, it consists of four concentric barrels and three disks per end-cap, for a total of more than 92 million pixels, each corresponding to an electronic channel [17]; the final space resolution is $10\ \mu\text{m}$ in the $R\text{-}\phi$ plane and $115\ \mu\text{m}$ along the z -axis;
- **semiconductor tracker** (SCT), it provides 8 measurements per track using four barrels and nine disks in the endcap regions based on silicon microstrip technology; the space resolution is $17\ \mu\text{m}$ in the $R\text{-}\phi$ plane and $570\ \mu\text{m}$ along

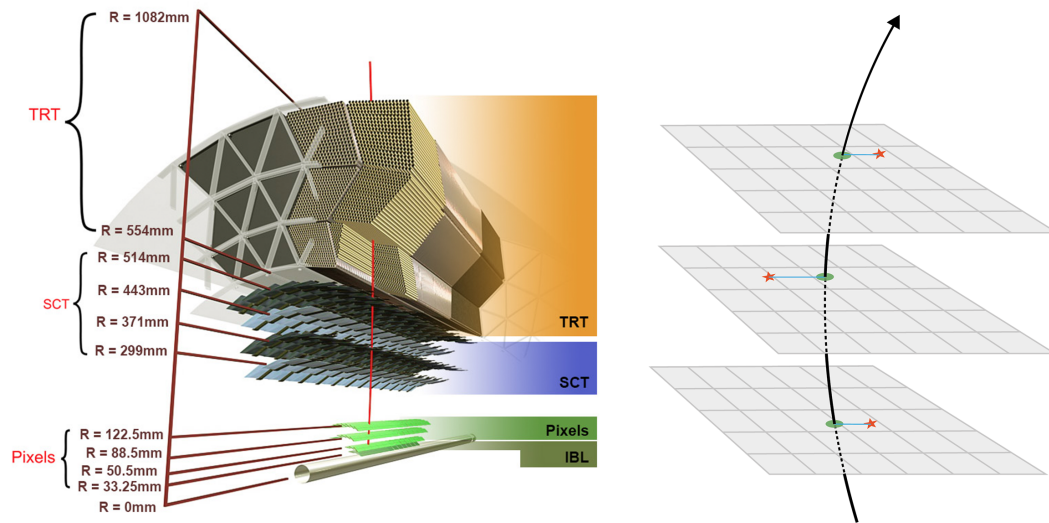


Figure 2.3: Layers of the ATLAS inner detector (left) and a schematic track reconstruction process (right) [16].

the z -axis.

A computer-generated image of the pixel tracker is shown in figure 2.4.

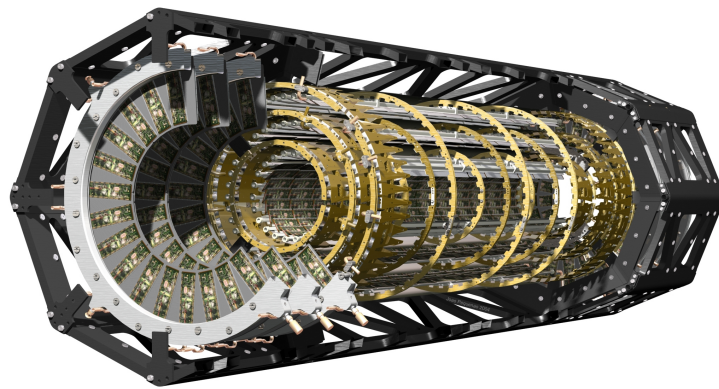


Figure 2.4: Computer-generated image of the ATLAS pixel tracker [18].

2.2.2 Transition Radiation Tracker

The *Transition Radiation Tracker* (TRT) is the outermost part of the ID and is based on gaseous detectors. More specifically, the detection technology uses both drift tubes and transition radiation, which is emitted when a charged particle

crosses the boundary between two materials with different refractive indices. This allows, for example, to efficiently distinguish electrons from pions in the range 1 GeV to 100 GeV. The TRT consists of 73 layers of straw tubes, each with a diameter of 4 mm, for a total of 350,000 channels. The space resolution is 130 μm per straw in both the R - ϕ z - ϕ directions.

2.2.3 Tracking Performance and the ITk

Using the hits from the three subdetectors, the ID measures the momentum of charged particles with the resolution reported in table 2.1, $\sigma_{p_T}/p_T = 0.05\% p_T \oplus 1\%$. The first term comes from the precision of the single hits and grows with p_T , since high-momentum tracks are less bent by the magnetic field and their curvature is harder to measure; the second one is due to multiple scattering in the detector material and does not depend on p_T . The two contributions are equal at $p_T \approx 20$ GeV, below which multiple scattering dominates, so the material in the ID (sensors, cables, cooling and support structures) has to be kept as low as possible.

The current ID was designed for the LHC design luminosity, corresponding to about 23 interactions per bunch crossing [13], and cannot be operated in the HL-LHC conditions described in section 1.3. With a pile-up of up to 200 the occupancy would be too high to reconstruct tracks efficiently, the readout links would not have enough bandwidth, and the radiation fluence¹ accumulated in the innermost layers over the HL-LHC program, above $2 \times 10^{16} n_{eq}/\text{cm}^2$, would exceed what the sensors can withstand. For these reasons, during the HL-LHC upgrade the whole ID will be replaced by the *Inner Tracker* (ITk) [19], an all-silicon detector without the TRT, extending the coverage up to $|\eta| < 4$, consisting of:

- **pixel detector**, with five barrel layers and several end-cap rings of $50 \mu\text{m} \times 50 \mu\text{m}$ pixels, for a total of about 13 m^2 of silicon and 5 billion channels; the innermost layers, receiving the highest dose, will be replaced after about 2000 fb^{-1} ;
- **strip detector**, with four barrel layers and six disks in each end-cap, for a total of about 160 m^2 of silicon.

In the forward region, the resolution on the longitudinal position of the tracks gets worse, making it noticeably harder to assign them to the correct vertex. To solve this problem, the ITk will be complemented by the *High-Granularity Timing*

¹The fluence is expressed in 1 MeV *neutron equivalents* (n_{eq}), i.e. the fluence of 1 MeV neutrons that would cause the same displacement damage in silicon as the actual mix of particles.

Detector (HGTD), placed in front of the end-cap calorimeters and covering the region $2.4 < |\eta| < 4.0$. Based on low gain avalanche detectors, it measures the time of each track with a resolution of 30 ps at the start of operation (then increasing to 50 ps towards the end due to radiation damage reducing the internal sensor gain), allowing to separate collisions that are close in space but occur at different times within the same bunch crossing [20].

2.3 The Calorimeter System

Calorimeters are used to measure the energy of particles by absorbing them and measuring the resulting showers of secondary particles. They are a core component of any high-energy physics experiment, providing fast real-time responses and being sensitive to both charged and neutral particles. Typically, their performance also improves with energy [13, 21]. The pseudorapidity coverage of the ATLAS calorimeter system is approximately $|\eta| < 4.9$; a cut away view of the calorimeters is shown in figure 2.5.

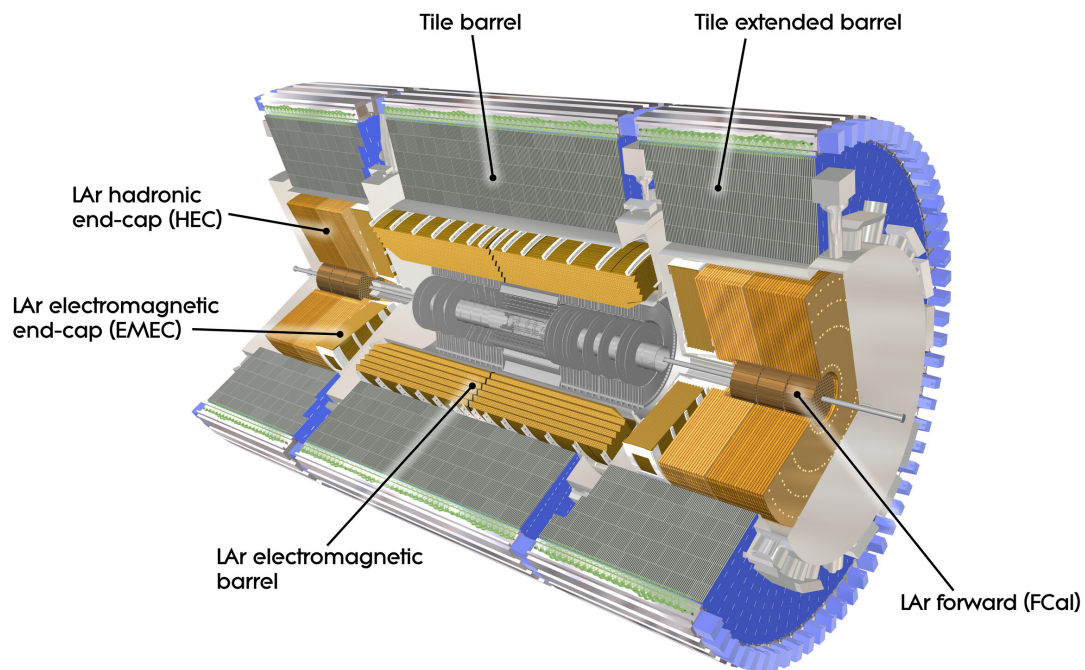


Figure 2.5: Cutaway of the ATLAS calorimeter system [22].

2.3.1 Electromagnetic Calorimeter

The *Liquid Argon Electromagnetic Calorimeter* (LAr EM) is a sampling calorimeter, made of lead absorbers and liquid argon as the active medium. Both absorbers and electrodes have an accordion shape over the full coverage, which provides complete symmetry in ϕ without azimuthal cracks. In the region $|\eta| < 1.8$ a *presampler*, consisting of a thin layer of liquid argon, is placed in front of the calorimeter to correct for the energy lost by electrons and photons before reaching it.

The calorimeter is divided into a barrel and two end-caps, covering different pseudorapidity ranges:

- **barrel**, covering $|\eta| < 1.475$, is made of three layers, with the first one having a very fine granularity in the η direction to distinguish between single photons and overlapping photons from neutral pion decays; the second layer collects most of the energy of the shower, while the third one samples its tail; its total thickness is more than $22 X_0$, X_0 being the radiation length; a schematic representation of the barrel can be found in figure 2.6;
- **end-caps**, covering $1.375 < |\eta| < 3.2$, each divided into an outer wheel ($1.375 < |\eta| < 2.5$) and an inner wheel ($2.5 < |\eta| < 3.2$); in the region $1.5 < |\eta| < 2.5$, matching the coverage of the inner detector, the structure follows the one of the barrel with three layers, while the inner wheel has only two layers and a coarser granularity; the total thickness is more than $24 X_0$.

2.3.2 Hadronic Calorimeter

The main requirement for hadronic calorimeters is to achieve efficient jet detection in the region $|\eta| < 5$ for jet tagging and missing transverse momentum reconstruction. The ATLAS hadronic calorimeter system consists of three different subdetectors, the *Tile Sampling Calorimeter* (Tile), *LAr Hadronic End-Cap Calorimeter* (HEC) and the *LAr Forward Calorimeter* (FCAL), covering a total of 10 mean nuclear interaction lengths (i.e. the mean distance a hadron covers before undergoing an inelastic nuclear interaction). Also, to make accurate single jet reconstruction, signature of many processes of interest, it's important to have good energy resolutions; the design values for the different subdetectors can be found in the ATLAS summary table 2.1.

The different subdetectors are optimized for different pseudorapidity ranges and use different technologies:

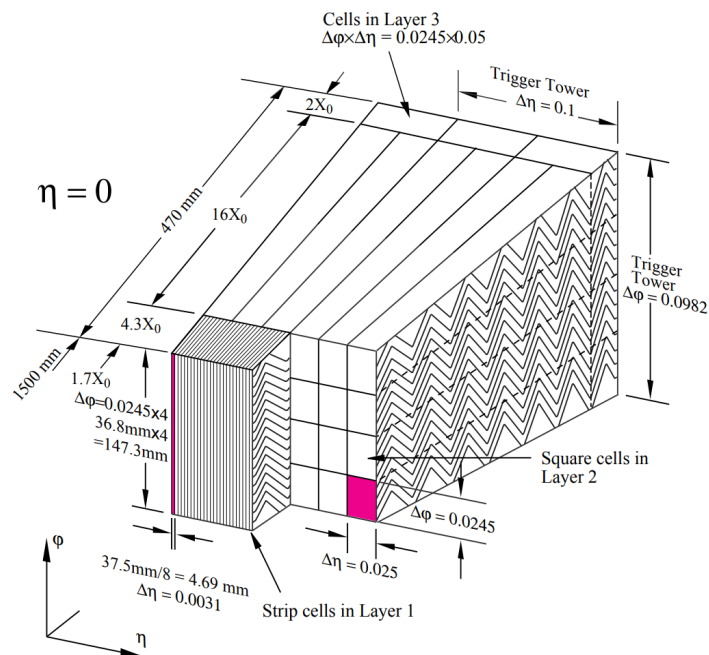


Figure 2.6: Schematic of the ATLAS electromagnetic calorimeter barrel [13].

- **Tile**, located in the barrel region ($|\eta| < 1.0$), is a sampling calorimeter with steel absorbers and scintillating tiles as the active medium; wavelength shifting fibers with photomultiplier tubes collect the light signal from the tiles;
- **HEC**, covering the region $1.5 < |\eta| < 3.2$, uses liquid argon as the active medium and copper as the absorber; it consists of two wheels per end-cap with 32 modules each;
- **FCAL**, covering the high particle flux region $3.1 < |\eta| < 4.9$, is a liquid argon calorimeter with copper absorbers in the first module and tungsten in the other two, for a total of three modules per end-cap; the first module is used mainly for electromagnetic measurements, while the other two leverage the better absorption properties of tungsten to measure hadronic showers.

2.3.3 Energy Resolution and Readout

The energy resolution of a calorimeter can be parametrized as:

$$\frac{\sigma_E}{E} = \frac{a}{\sqrt{E}} \oplus \frac{b}{E} \oplus c \quad (2.1)$$

where:

- a is the *stochastic term*, due to the fluctuations in the development of the shower and in the fraction of energy deposited in the active medium;
- b is the *noise term*, due to the electronic noise and to the energy deposited by pile-up;
- c is the *constant term*, due to the non-uniformities of the detector and of its calibration.

Contrary to the tracking, the relative resolution improves with energy: using the design values of table 2.1, which do not include the noise term, the constant term dominates above about 200 GeV for electromagnetic showers and 280 GeV for jets, while the noise term becomes relevant only for low-energy deposits.

In the LAr barrel, the ionization electrons take about 450 ns to drift across the liquid argon gap, much more than the 25 ns bunch spacing. The signal of a deposit therefore overlaps with the ones from the following bunch crossings, making the LAr calorimeters strongly affected by the out-of-time pile-up described in section 1.2.2. To limit this effect, the front-end electronics shape the signal into a bipolar pulse, with the positive and negative lobes having the same area: this way, on average, the contributions from the previous crossings cancel out. The pulse is then sampled every 25 ns, and five samples around the peak are used to reconstruct the energy and time of the deposit [13].

For the HL-LHC, the readout electronics of both the LAr and Tile calorimeters will be replaced, since the current ones cannot withstand the expected radiation doses nor meet the requirements of the new trigger system [23]. The new electronics will digitize the signals of every calorimeter cell at each bunch crossing and send them off the detector, making the full-granularity information available to the trigger.

2.4 The Muon Spectrometer

The muon spectrometer is the outermost part of the ATLAS detector, surrounding the hadronic calorimeters with 8 coils in the barrel and two end-caps. Its main purpose is to precisely measure the muon momentum in the region $|\eta| < 2.7$ and to provide a fast muon trigger (for $|\eta| < 2.4$): for this reason, different detector technologies are used depending on the specific requirements of the subdetector. A schematic of the muon spectrometer is shown in figure 2.7.

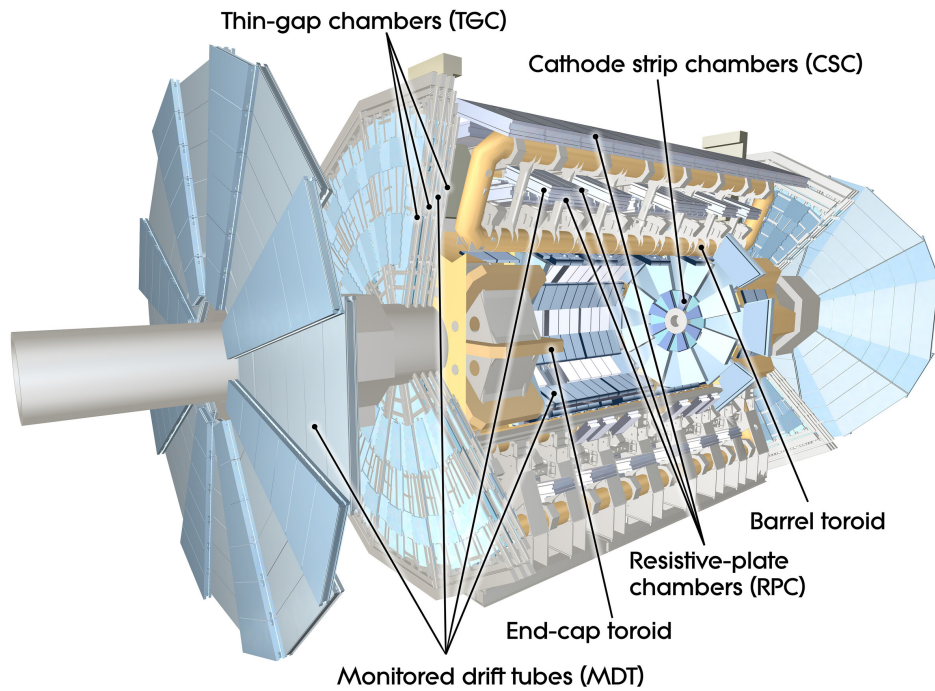


Figure 2.7: Cutaway of the ATLAS muon spectrometer [24].

2.4.1 Precision Chambers

Two different technologies are implemented for the precision measurements of the muon tracks:

- **Monitored Drift Tubes (MDTs)**, covering most of the spectrometer's surface, they fully span the region $|\eta| < 2.7$ and implement single-wire drift tube technology using aluminum tubes filled with Ar/CO_2 . When a muon passes through the tube, it ionizes the gas, producing electrons; those drift towards the tungsten/rhenium wire, producing secondary ionizations and an avalanche, which makes the signal detectable. Knowing the drift velocity of the electrons, the distance of the muon from the wire can be calculated from the time it takes for the signal to be produced (i.e. for the electron to reach the wire). A schematic representation of the MDTs is shown in figure 2.8.
- **Cathode Strip Chambers (CSCs)** are multi-wire proportional chambers placed in the innermost part of the end-caps, covering the region $2.0 < |\eta| < 2.7$. They can sustain much higher rates than MDTs, hence they were chosen to instrument the region with the highest particle flux. Cathode strips are

placed both perpendicular and parallel to the wires, allowing to measure the position of the muon in two coordinates.

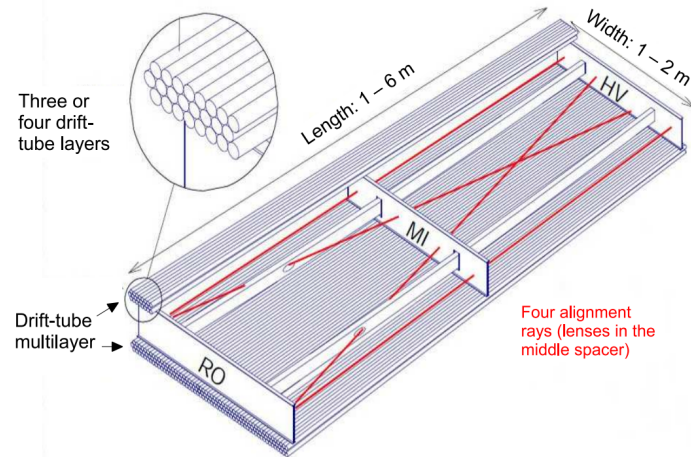


Figure 2.8: Schematic of an ATLAS MDT chamber [13].

2.4.2 Trigger Chambers

The focus of trigger chambers is to provide a fast response, much more important for a trigger system than spatial resolution. Two different technologies are used:

- **Resistive Plate Chambers (RPCs)** are gaseous detectors used in the barrel region ($|\eta| < 1.05$); their time resolution of about 1.5 ns makes them ideal for the trigger system, compromising on the spatial resolution (of about 1 cm).
- **Thin Gap Chambers (TGCs)** are multi-wire proportional chambers placed in the region $1.05 < |\eta| < 2.7$, used for triggering up to $|\eta| = 2.4$; they have a time resolution of about 4 ns and a spatial resolution between 2 mm and 7 mm.

Both types of chambers also measure the coordinate of the track in the non-bending plane, which is not measured by the MDTs and is therefore used as the second coordinate of the MDT measurement [13].

2.4.3 New Small Wheels

During the second Long Shutdown (2019–2022), the inner station of the muon end-caps, the so-called *Small Wheel*, including the CSCs described above, was

replaced by the *New Small Wheels* (NSWs), covering the region $1.3 < |\eta| < 2.7$ [25]. The main reason for this upgrade was the muon trigger in the end-caps: about 90 % of the single muon triggers did not correspond to a real muon coming from the interaction point, but to background particles produced in the material between the inner and middle stations. Since the trigger decision was taken using only the TGCs of the middle station, there was no way to check whether the track actually came from the interaction point. With the NSWs, a muon candidate is accepted only if a matching track segment pointing to the interaction point is also found in the inner station, thus rejecting most of the fake triggers.

Each NSW is a disk of about 10 m in diameter, divided into 8 large and 8 small sectors, and combines two detector technologies:

- **small-strip Thin Gap Chambers** (sTGCs), an evolution of the TGCs with finer readout strips, mainly used for the trigger thanks to their timing performance;
- **Micromegas** (MMs), gaseous detectors in which the ionization electrons drift through a thin mesh into a narrow amplification gap, mainly used for precision tracking since they can sustain high particle rates.

Each sector is made of two Micromegas wedges placed between two sTGC wedges, each wedge having four detector layers. Both technologies contribute to the trigger and to the tracking, with a spatial resolution better than $100\text{ }\mu\text{m}$ per layer, needed to reconstruct the track segment with the angular resolution of 1 mrad required by the trigger.

2.5 Forward Detectors and Luminosity Measurement

Besides the main subsystems, ATLAS includes four smaller detectors placed in the forward region, along the beam line and at large distances from the interaction point. Their layout is shown in figure 2.9.

2.5.1 Forward Detectors

Forward detectors are placed symmetrically on both sides of the interaction point:

- **LUCID-2** (*L*uminosity *C*herenkov *I*ntegrating *D*etector), located at about 17 m from the interaction point. It consists of 16 photomultipliers per side placed around the beam pipe, with their quartz windows acting as Cherenkov

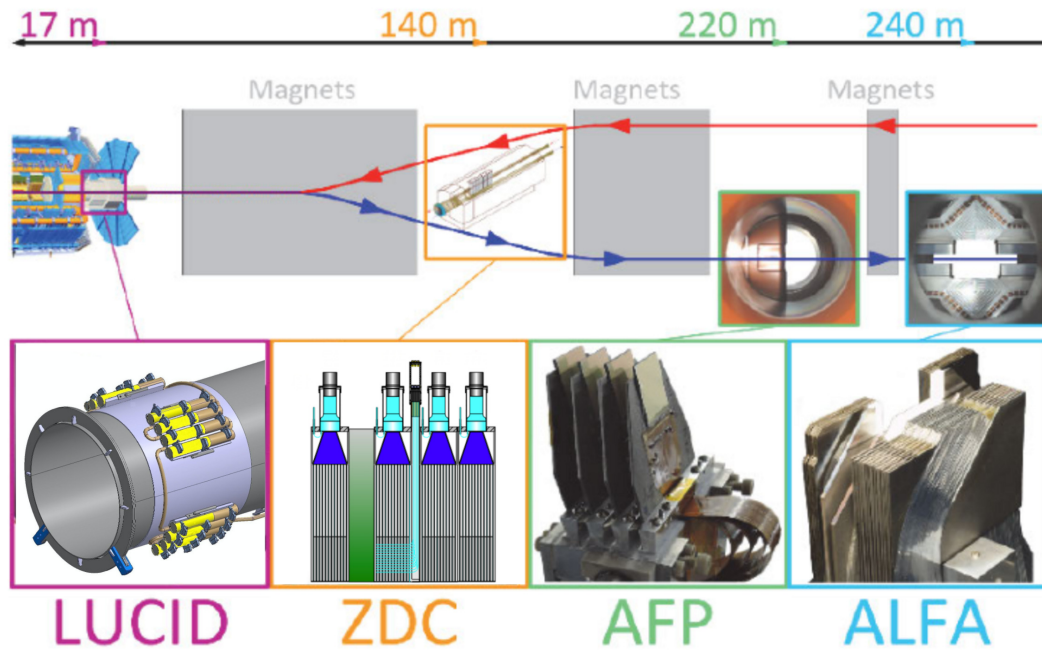


Figure 2.9: Layout of the ATLAS forward detectors [26].

radiators for the charged particles produced in the collisions. It covers the role of ATLAS main luminosity monitor.

- **ZDC** (*Zero Degree Calorimeters*), at about 140 m, just after the two beams are separated into different beam pipes. Made of alternating quartz rods and tungsten plates, it measures the neutral particles at $|\eta| \geq 8.2$ (mainly spectator neutrons) and it's used to determine heavy-ion collision centrality.
- **AFP** (*ATLAS Forward Proton*), installed in 2015 and 2016, with stations at about 205 m and 217 m. Using a silicon pixel tracker and a time-of-flight detector, it measures the protons scattered at very small angles in diffractive events, mainly during special runs at low pile-up.
- **ALFA** (*Absolute Luminosity For ATLAS*), at about 240 m, made of scintillating fiber trackers housed in *Roman pots*, special movable devices inserted in the beam pipe that can approach the beam down to 1 mm. It is used in special runs with high β^* to measure the elastic scattering at very small angles, from which the total proton-proton cross-section is obtained.

2.5.2 Luminosity Measurement

The luminosity values introduced in section 1.2.3 are measured by ATLAS itself, mainly using LUCID-2, the only ATLAS luminometer able to measure the absolute luminosity of each bunch under all beam conditions. For each bunch crossing, a *hit* is recorded when the signal of a photomultiplier exceeds a given threshold. Similarly to (1.7), the average number of interactions per bunch crossing seen by the detector is:

$$\mu_{vis} = \frac{\sigma_{vis} \mathcal{L}_{BC}}{f_{rev}} \quad (2.2)$$

where $\mathcal{L}_{BC}$ is the luminosity of a single pair of colliding bunches and σ_{vis} the *visible cross-section*, i.e. the inelastic cross-section multiplied by the efficiency of the detector in recording a hit. Since the number of interactions follows a Poisson distribution, the fraction f of bunch crossings with at least one hit is $f = 1 - e^{-\mu_{vis}}$, giving:

$$\mathcal{L}_{BC} = -\ln(1 - f) \frac{f_{rev}}{\sigma_{vis}} \quad (2.3)$$

The only unknown is σ_{vis} , which is calibrated with *van der Meer* scans [27]: in dedicated fills at low luminosity ($\mathcal{L} \sim 10^{31} \text{ cm}^{-2} \text{ s}^{-1}$), the two beams are moved across each other in the horizontal and vertical directions, and the rate measured as a function of their separation gives the transverse size of the overlap region. Knowing the number of protons per bunch, measured by the LHC, it's possible to determine the absolute luminosity, from which σ_{vis} follows. The calibration is then extrapolated to the physics conditions, with a much higher pile-up, and monitored throughout the year by comparing LUCID-2 with other independent measurements, such as the number of tracks reconstructed in the ID or the currents in the LAr and Tile calorimeters.

The luminosity measured *online* is sent to the LHC every 1-2 seconds, with an accuracy of about 5 %; this information is mainly used by operators to monitor and optimize the beams. The final offline measurement is much more precise though: for the full Run 2 dataset at $\sqrt{s} = 13 \text{ TeV}$ the total uncertainty is 0.83 % [27]. For the HL-LHC, LUCID-2 will be replaced by an upgraded version, LUCID-3, whose prototypes were already installed during Run 3 [26].

2.6 Object Reconstruction

The signals recorded by the subdetectors have to be combined to reconstruct the physics objects used in the analyses, such as jets, electrons, photons, muons and

missing transverse momentum. The same reconstruction algorithms are applied both to data and to simulated events, so that the two can be directly compared.

Most objects rely on the tracks of charged particles reconstructed in the ID. Starting from seeds made of three space points in the silicon detectors, the track candidates are extended through the remaining layers with a Kalman filter, and the ambiguities between tracks sharing the same hits are solved based on a quality score [28]. The tracks are then used to reconstruct the interaction vertices, each with at least two associated tracks: among them, the *primary vertex* of the hard-scatter interaction is chosen as the one with the highest $\sum p_T^2$ of the associated tracks, while the others are considered as pile-up vertices [29].

2.6.1 Jets

Quarks and gluons produced in the collisions cannot be observed directly: due to color confinement they hadronize, producing a collimated spray of hadrons called *jet*. Jets are reconstructed in two steps: first the calorimeter cells are grouped into clusters, then the clusters are combined by a jet algorithm.

The clusters, called *topological clusters* (topo-clusters), are built based on the significance of the energy of each cell with respect to its expected noise σ_{noise} , which includes both the electronic and the pile-up noise [30]. More specifically:

- cells with $|E| > 4\sigma_{noise}$ are used as seeds;
- neighboring cells with $|E| > 2\sigma_{noise}$ are added iteratively, extending the search to their neighbors as well;
- finally, all the cells adjacent to the cluster are added, regardless of their energy.

Clusters with more than one local maximum are then split, to separate the showers of nearby particles.

The clusters are then combined with the *anti- k_t* algorithm [31], part of a family of sequential recombination algorithms based on the distances:

$$d_{ij} = \min\left(p_{T,i}^{2p}, p_{T,j}^{2p}\right) \frac{\Delta R_{ij}^2}{R^2} \quad d_{iB} = p_{T,i}^{2p} \quad (2.4)$$

where d_{ij} is the distance between two objects i and j , d_{iB} the distance between the object i and the beam, ΔR_{ij} their angular separation as defined in section 2.1.1 and R the radius parameter. At each step the smallest distance is found: if it is a

d_{ij} the two objects are merged, while if it is a d_{iB} the object is declared a jet and removed from the list. The value of p defines the algorithm: $p = 1$ gives the k_t algorithm, $p = 0$ the Cambridge/Aachen and $p = -1$ the anti- k_t .

With $p = -1$, soft particles are clustered with the hardest ones before being merged with each other, producing jets with an approximately conical shape of radius R , only slightly affected by soft radiation and pile-up. Like the others in the family, the algorithm is also *infrared and collinear safe*, i.e. the resulting jets do not change if a soft emission or a collinear splitting is added to the event. In ATLAS the standard choice for jets coming from single quarks or gluons is $R = 0.4$. When a heavy particle is produced with a large transverse momentum, however, its decay products are collimated: since their angular separation is approximately $\Delta R \approx 2m/p_T$, with m the mass of the particle, for $p_T \gtrsim 2m$ they are both contained in a jet of radius $R = 1.0$. In this *boosted* regime, *large-R* jets are used instead, so that the whole decay is reconstructed as a single jet. The total jet mass can then be used to identify the original particle, since both its decay products are contained in the same *boosted jet*.

Having a much larger area, large-R jets also collect more energy from pile-up and soft radiation, which degrades the resolution on their mass. For this reason, *grooming* techniques are applied, removing from the jet the soft and wide-angle components that are not related to the decay of the original particle. Several grooming algorithms exist, differing in the criteria used to identify these components, such as trimming [32] or soft drop [33].

Instead of using only calorimeter clusters, the jet inputs can also be built with the *particle flow* algorithm [34], which combines them with the tracks reconstructed in the ID: for each track, the expected energy deposited in the calorimeter is subtracted from the matching clusters and replaced by the track momentum. This improves the resolution at low p_T , where the tracker is more precise than the calorimeter, and allows to remove the charged particles coming from pile-up vertices.

The energy measured for a reconstructed jet is generally different from the energy of the particles that form it. The ATLAS calorimeters are *non-compensating*, i.e. hadrons produce a smaller signal than electrons and photons of the same energy; moreover, part of the energy is deposited in inactive material, such as cables and supports, or falls below the noise thresholds used to build the topo-clusters, while pile-up adds energy that does not belong to the jet. The *jet energy scale* (JES) is therefore corrected in several steps [35]:

- **pile-up correction**, subtracting the pile-up contribution based on the jet

area and on the average energy density of the event;

- **absolute calibration**, correcting the jet energy and direction to the ones of particle-level jets in simulation;
- **global sequential calibration**, reducing the dependence of the response on the jet structure, such as the fraction of energy in each calorimeter layer or the number of associated tracks;
- **in-situ calibration**, applied to data only, correcting the residual differences with the simulation by balancing the jet against well-measured objects, such as photons or Z bosons.

The *jet energy resolution* (JER) is measured with similar techniques. Jets coming from pile-up interactions can finally be rejected with the *Jet Vertex Tagger* (JVT), a discriminant based on the fraction of the jet tracks associated with the primary vertex [36].

2.6.2 Other Physics Objects

Besides jets, the main objects reconstructed in ATLAS are:

- **electrons**, reconstructed from clusters in the EM calorimeter matched to a track in the ID, and identified with a likelihood discriminant based on the shower shape, the track quality and the TRT information [37];
- **photons**, reconstructed from EM clusters with no matching track (*unconverted*) or matched to a conversion vertex, if the photon converted into an electron-positron pair in the ID material (*converted*) [37];
- **muons**, mainly reconstructed by combining a track in the ID with a track in the muon spectrometer [38];
- **hadronic τ decays**, reconstructed as narrow jets with one or three associated tracks;
- **b -jets**, jets originating from b -quarks, identified by exploiting the long lifetime of b -hadrons, which produces secondary vertices “displaced” from the primary one, measured leveraging the pixel detector high precision;
- **missing transverse momentum** p_T^{miss} , defined as the negative vector sum of the transverse momenta of all the reconstructed objects, plus a *soft term* built from the tracks not associated with any of them [39].

2.6.3 Simulation

Simulated events are needed to design and interpret the analyses, comparing experimental data with predictions of the SM and of new physics models. The ATLAS simulation chain consists of several steps [40]:

- **event generation**, where a Monte Carlo generator produces the particles from the hard scattering, followed by the parton shower, the hadronization and the decay of unstable particles;
- **detector simulation**, where the interactions of the generated particles with the detector material are simulated with GEANT4 [41], or with faster parametrized methods for the calorimeters;
- **digitization**, where the energy deposits are converted into detector signals, including the electronics response and pile-up, added by overlaying simulated minimum-bias interactions.

The output has the same format as real data and is processed with the same reconstruction algorithms described above.

The information on the generated particles is also kept, allowing to define *truth* (or particle-level) objects: truth jets, for example, are built with the anti- k_t algorithm directly from the stable generated particles ($c\tau > 10$ mm), excluding muons and neutrinos, and are the reference used in the absolute jet calibration.

Chapter 3

The ATLAS Trigger and Data Acquisition System

Only a small fraction of the collisions produced at the LHC can be recorded: the choice of which events to keep, and the transport of their data from the detector to permanent storage, are the tasks of the trigger and data acquisition system. The work presented in this thesis concerns this system and its upgrade for the HL-LHC: this chapter introduces the general principles of trigger and data acquisition, and describes the ATLAS system in Run 3 and its evolution for the HL phase.

3.1 Trigger and Data Acquisition Principles

In a High Energy Physics experiment, the *Trigger and Data Acquisition* (TDAQ) system manages the selection of events of physics interest to be kept for further analysis. To do so, a complex system of hardware and software components processes the detector signals, operates the selection and organizes selected events in a format suitable for storage and offline analysis. This selection process can be summarized in two main phases:

- **Trigger:** the trigger system operates the selection. It is designed to have a predictable decision time (fixed, in the case of the first level) to avoid losing events (section 3.1.1), therefore normally simple criteria are implemented (at least regarding the first levels, operating at the highest rate). The trigger criteria are listed in the *trigger menu*, a set of trigger chains required to cover all the physics channels of interest as well as other data samples needed for calibration and monitoring purposes.

- **Data Acquisition:** after the event is effectively selected, the data acquisition system is responsible for collecting event fragments, assembling them into complete events, and sending them to storage for offline analysis.

The requirements of the two phases are partially in conflict. The selection has to keep the events of interest with a high and well-known efficiency, since any loss or bias directly affects the physics measurements, such as the cross-sections; at the same time, it has to reject enough of the much more frequent background processes to fit within the bandwidth of the data acquisition system and the storage and computing resources available offline. All of this has to be done within a limited processing time, which sets how complex the selection algorithms can be. The following sections describe the main concepts used to address these requirements: dead time and buffering, pipelined and multi-level triggers, and event building.

3.1.1 Dead Time and Buffering

During the processing phase, a DAQ system typically isn't ready to receive new events (it's *busy*): this time window is referred to as *dead time*. Several factors contribute to dead time: *trigger latency* (the time between the event occurrence and the trigger decision), *detector latency*, operational timeouts and, in general, any form of delay. Therefore, when designing a DAQ system, the goal is to minimize the dead time, keeping in mind that it would be impossible to eliminate it completely and accounting for it when evaluating the overall efficiency of the system.

Assuming a stochastic process and a constant dead time, the *data acquisition rate* ν can be defined. More specifically, if the probability of the system being busy is

$$P[busy] = \nu\tau$$

where τ is the dead time, then the probability of it being free to accept a new event is

$$P[free] = 1 - \nu\tau$$

Assuming a physical process with frequency ν_p , the data acquisition rate can be expressed as

$$\nu = \nu_p P[free] = \nu_p (1 - \nu\tau)$$

which can be solved to give

$$\nu = \frac{\nu_p}{1 + \nu_p\tau}$$

Furthermore, it's possible to define the *efficiency* of the DAQ system, corresponding to the ratio between the number of acquired events and the total number of events produced by the physical process:

$$\epsilon = \frac{N_{\text{saved}}}{N_{\text{tot}}} = \frac{\nu}{\nu_p} = \frac{1}{1 + \nu_p \tau} < 100\% \quad (3.1)$$

In figure 3.1 the output frequency ν and the efficiency ϵ are shown as a function of the input frequency ν_p . It's evident that the efficiency drops steeply with the input frequency, meaning that to maintain optimal efficiency levels the dead time should be kept extremely low $\tau \ll 1/\nu_p$.

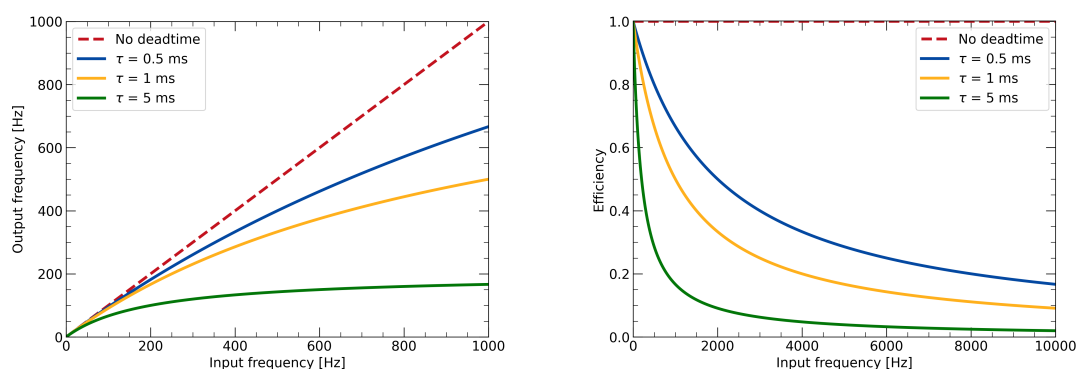


Figure 3.1: Output frequency ν (left) and efficiency ϵ (right) as a function of the input frequency ν_p , for different values of the dead time τ .

As a numerical example, a physical process with $\nu_p = 1$ kHz and a DAQ dead time $\tau = 1$ ms gives an acquisition rate $\nu = 500$ Hz, i.e. an efficiency of only 50 %. To reach an efficiency of 99 %, (3.1) requires $\tau \lesssim 10$ μ s: the system has to be able to process events at 100 kHz, two orders of magnitude more than the average rate of the process. Since the events arrive at random times, this over-design is needed to avoid losing the ones that come too close to each other, which is clearly not sustainable; a better solution is to absorb these fluctuations with buffers.

A buffer, or *queue*, placed before the processing stage stores the events arriving while the system is busy, so that they can be processed later instead of being lost: the processing stage then only has to keep up with the average rate of the events, and not with their fluctuations. The behavior of such systems is described by *queuing theory*.

Generally speaking, queues are characterized by the stochastic properties of the *inter arrival time* of requests, as their distribution, assumed to be random, independent and identically distributed variables. Other important parameters are the

service time (the time needed to process a request), the total number of servers, the system capacity (determining the maximum number of customers in the queuing systems) and the *service discipline*, determining the order in which requests are served.

The most common service disciplines are:

- *First in First Out* (FIFO): the first request to arrive is the first to be served (only one at a time);
- *Last in First Out* (LIFO): the last request to arrive is the first to be served (only one at a time);
- *Priority Service* (PS): requests are served according to their assigned priority, with the highest priority being served first;
- *Random Service* (RS): requests are served in a random order.

The overall performance of a queuing system can be evaluated introducing the *traffic intensity* ρ :

$$\rho = \frac{\text{mean service time}}{\text{mean inter arrival time}} = \text{arrival density} \times \text{mean service time}$$

A $\rho > 1$ would refer to an overloaded system, where new servers would be needed to keep up with the incoming requests. In figure 3.2 the efficiency of a FIFO buffer against the traffic intensity is shown, for different buffer depths (where unitary depth correspond to a no-buffer scenario). A noticeable improvement in efficiency is observed already with a buffer depth of 5; further increasing the depth gives diminishing returns, an important characteristic to be considered when designing a DAQ system with limited resources. Typically, the smallest buffer depth guaranteeing optimal efficiency is chosen, to minimize the cost and complexity of the system.

3.1.2 Pipelined and Multi-Level Triggers

At the LHC a new bunch crossing occurs every 25 ns, a much shorter time than the one needed by any trigger to take a decision; this means that a trigger system that waits for the decision before accepting the next crossing would have a huge dead time, losing numerous events. This problem is solved by *pipelining*: the data of each bunch crossing are stored in the front-end electronics in *pipeline memories*, while the trigger processes them in several consecutive steps, working on many crossings at the same time. The decision for each crossing is then available after

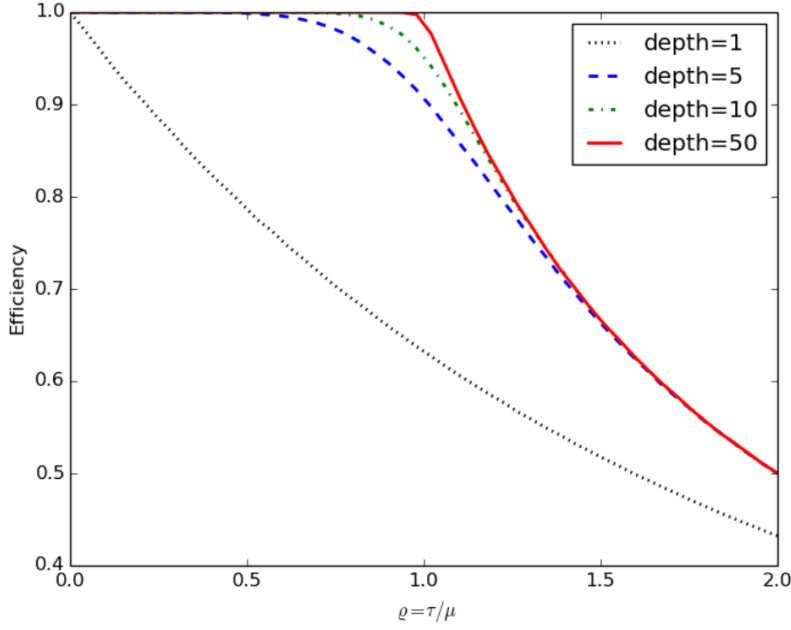


Figure 3.2: Efficiency of a FIFO buffer as a function of the traffic intensity ρ , for different buffer depths [42].

a fixed time, the *latency*, after which the data are either read out, if the event is accepted, or overwritten. In this way the latency only sets the depth of the pipeline memories, which have to store all the crossings for which a decision is still pending: with a latency of 2.5 μ s, for example, 100 crossings.

The data of the accepted events are then moved from the pipeline memories to *derandomizing buffers*, which are read out at a constant rate. As described in section 3.1.1, these buffers absorb the fluctuations of the trigger rate, and the dead time is limited to the cases in which they are full; to avoid this, the trigger can apply some rules limiting the number of events accepted in a short time.

Even with pipelining, performing the entire selection on every bunch crossing would require too much time and processing power. The solution is to split the selection process into several levels, creating a *multi-level trigger*. The first level operates a rough but fast selection, using only simple criteria and a reduced amount of detector information, and is usually implemented in custom hardware. Since each level reduces the rate of events passed to the following one, the time available to process each event increases at each step, allowing for more complex and precise selection criteria, up to the last levels, usually implemented in software on commercial computers.

To reduce the amount of data to be processed, the first level also identifies the *Regions of Interest* (RoIs), i.e. the regions of the detector where the candidate objects were found: the following levels can then analyze only the data in these regions, accessing the full event only when needed. At the last levels, the selection algorithms are kept as close as possible to the ones used in the offline reconstruction, so that the events are selected with the same definitions of the physics objects later used in the analyses. The rate reduction achieved at each level, together with the time available to take the decision, is shown in figure 3.3.

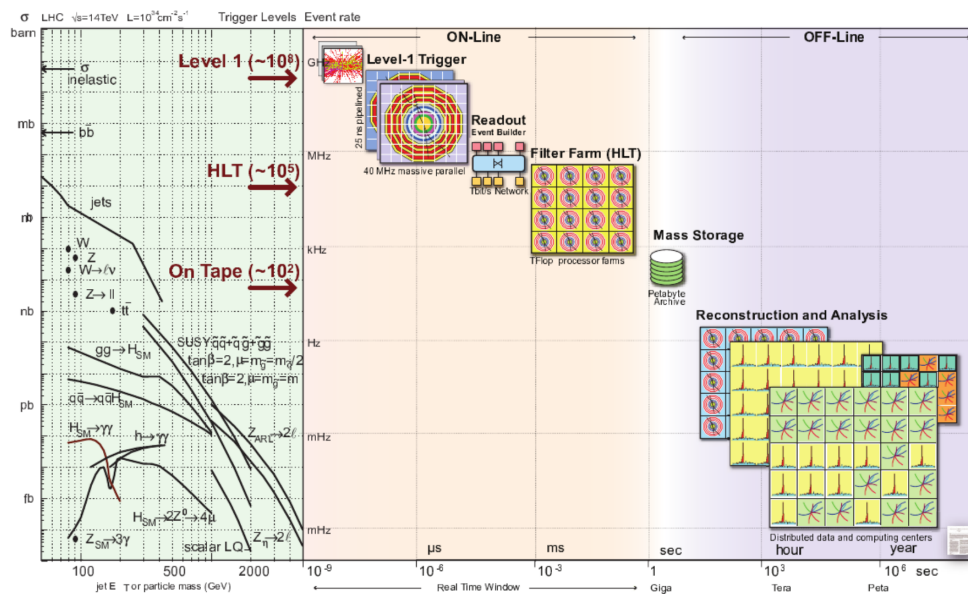


Figure 3.3: Event rates and decision times at the different levels of a trigger and data acquisition system [43].

3.1.3 Event Building

The data of a single event are produced by many different readout channels, distributed across the whole detector: each of them sends a *fragment*, containing only its part of the event and labelled with an identifier distributed by the trigger system to all the detector components. The *event building* process consists in collecting all the fragments with the same identifier and assembling them into a single, complete event, which can then be processed by the higher trigger levels or sent to storage.

The way the data sources and the processing units are connected to each other is defined by the *readout topology*, of which two main types exist, shown in figure 3.4:

- **bus topology**, where all the devices are connected to the same shared transmission line, the *bus*, and receive all the signals travelling on it, ignoring the ones not addressed to them; since only one device at a time can send data on the bus, a master/slave relationship is needed to organize the communication. The main advantage of this topology is its simplicity, with a well-defined number of lines and well-defined protocols. The major drawback, on the other hand, is its limited scalability: the bandwidth is shared among all the connected devices, the width of the bus is fixed, and both its frequency and the number of devices it can serve decrease with its length;
- **network topology**, where the devices communicate directly with each other through *switches*, which route the messages from their sources to their destinations. No arbitration is needed, and different devices can communicate at the same time. Switches also have to handle *congestion*, i.e. multiple messages trying to reach the same destination at the same time: data loss is avoided using higher-level communication protocols and buffers, which are placed throughout the network to smooth out the data flow.

For large experiments the network topology scales much better, and it is the one used in the ATLAS dataflow since Run 1.

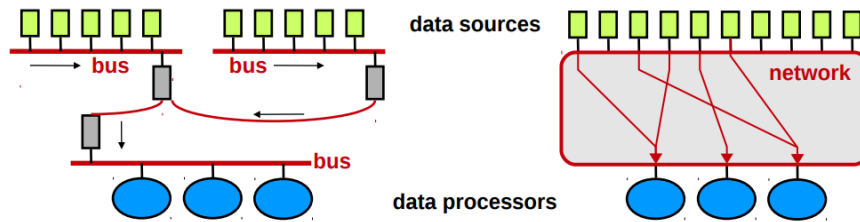


Figure 3.4: Schematic representation of bus and network topologies [44].

In a network-based system, the transfer of the fragments can be organized in two ways: in a *push* scheme, the sources send their fragments as soon as they are available to a destination, typically assigned by a supervising component; in a *pull* scheme, instead, the destination requests the fragments from the sources only when it needs them. In both cases, all the fragments of an event are sent to the same destination at about the same time, and the corresponding switch port can be overloaded, causing congestion and possibly data loss. This effect can be mitigated with large buffers in the switches, or with *traffic shaping* techniques, such as limiting the number of fragments requested at the same time by each destination. The way these concepts are implemented in the ATLAS TDAQ system is described in sections 3.3 and 3.4.

3.2 Requirements

As for many High Energy Physics experiments, the TDAQ system of the ATLAS experiment is designed as a two stage selection system, with the first stage being a hardware-based trigger (*Level-1*) and the second stage being a software-based trigger (*High-Level Trigger* or *HLT*). Following the principles described in section 3.1, the Level-1 trigger reduces the rate from the 40 MHz of the bunch crossings to 100 kHz, while the HLT further reduces it to a few kHz, the rate at which events can be saved to storage.

The main parameters of the system are summarized in table 3.1, both for Run 3 and for the HL-LHC. The higher pile-up expected at the HL-LHC, up to 200, results in much larger events, and in order to keep the trigger thresholds low enough for the physics program, the Level-1 accept rate of the first trigger level, called *Level-0* (L0) in the HL-LHC upgraded system, has to increase by a factor of ten, reaching 1 MHz. As a consequence, the readout system has to sustain a throughput more than twenty times larger than in Run 3, while the rate of events saved to storage increases to 10 kHz.

At the same time, the first trigger level latency increases from 2.5 μs to 10 μs , to allow more complex selection algorithms: as described in section 3.1.2, this requires pipeline memories able to store 400 bunch crossings instead of 100, one of the reasons for the replacement of the front-end electronics described in chapter 2.

Table 3.1: Main parameters of the ATLAS TDAQ system in Run 3 and after the HL-LHC upgrade [26, 45, 46].

	Run 3	HL-LHC
Bunch crossing rate	40 MHz	40 MHz
Pile-up $\langle\mu\rangle$	< 60	< 200
L1/L0 accept rate	100 kHz	1 MHz
L1/L0 latency	2.5 μs	10 μs
Event size	2.1 MB	4.6 MB
Readout throughput	0.2 TB/s	4.6 TB/s
Output rate to storage	3 kHz	10 kHz

3.3 The Run 3 TDAQ System

The TDAQ system used during Run 3 is shown in figure 3.5. The Level-1 trigger uses reduced-granularity information from the calorimeters and the muon spectrometer to select events at a rate of up to 100 kHz; for each accepted event, the data are read out from the front-end electronics and buffered in the readout system, from where they are requested by the HLT. The events accepted by the HLT are finally sent to permanent storage, at an average rate of about 3 kHz [26]. The following sections describe the Level-1 trigger, the readout and dataflow system, and the HLT with its selection menu.

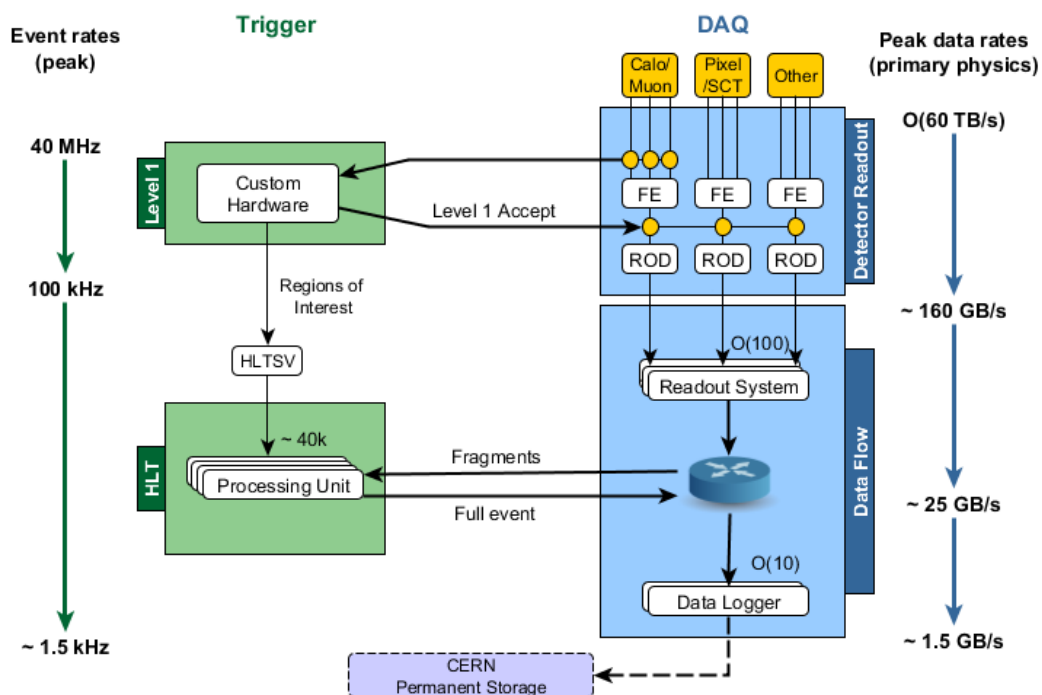


Figure 3.5: Schematic of the ATLAS TDAQ system in Run 3 [26].

3.3.1 The Level-1 Trigger

The Level-1 trigger represents the first, hardware-based stage of the ATLAS event selection chain. Its role is to reduce the event rate from 40 MHz to 100 kHz, i.e. to select about 0.25 % of the bunch crossings. Each decision is performed within a fixed latency of 2.5 μs , where various logic algorithms are applied to the

reduced-granularity information from the calorimeters and the muon spectrometer. More specifically, the Level-1 trigger identifies from the different L1 subsystems the events containing photon, jet, electron, muon or hadronically decaying tau candidates, as well as events with a large E_T^{miss} . Finally, the *Central Trigger Processor* (CTP) combines the information from the different subsystems and computes the final *L1-Accept* (L1A) decision, sent to the front-end electronics to read out the event data.

L1 Calorimeter Trigger

Before the start of Run 3, the Level-1 Calorimeter trigger (L1Calo) was upgraded to receive finer-granularity information from the LAr calorimeters, now digitized directly in the LAr front-end electronics. While in Run 2 the inputs consisted of *trigger towers* of 0.1×0.1 in η - ϕ , the electromagnetic information is now provided as *SuperCells*, sums of four or eight calorimeter cells, with typically ten SuperCells for each trigger tower (for example, for $|\eta| < 1.4$, a single SuperCell covering the whole 0.1×0.1 tower in the presampler and in the back layer, and four SuperCells of 0.025×0.1 in each of the front and middle layers) [26]. The Tile calorimeter data are instead still received in analog form, digitized by the *Preprocessor* and sent to the new system by the *Tile Rear Extension* (TREX) modules [47]. These inputs are processed by three new *Feature Extractors* (FEXs):

- **electromagnetic FEX** (eFEX): uses the full SuperCell granularity to reconstruct electron, photon and hadronically decaying tau candidates, together with the shower-shape variables used for their identification;
- **jet FEX** (jFEX): reconstructs jets with $R = 0.4$ and $R = 0.8$, tau candidates and forward electromagnetic objects, and computes the E_T^{miss} , using towers of 0.1×0.1 in the central region;
- **global FEX** (gFEX): processes the data from the entire calorimeter on a single module, with a coarser granularity, to identify large- R jets and compute global quantities such as the E_T^{miss} .

The Run-2 system, now referred to as *legacy*, can be operated in parallel to the upgraded one: its *Cluster Processor* and *Jet/Energy-sum Processor* identify, respectively, electron, photon and tau candidates, and jet candidates and global energy sums. In 2022 the calorimeter-based physics triggers were still provided by the legacy system, while the new electromagnetic objects were commissioned to be used from the start of the 2023 data taking [47].

After the above operations, results are sent to the CTP for the computation of the

final L1A decision. In case the event is accepted, coordinates of the identified candidates are sent to the HLT as *Regions of Interest* (RoIs).

L1 Muon Trigger

The Level-1 Muon trigger (L1Muon) is based on the information coming from both the RPC and TGC detectors. Muon candidates are identified by comparing the pattern of hits in the trigger chambers with the one expected for a muon of infinite momentum: the deviation from this pattern gives an estimate of the transverse momentum, which is compared with six programmable thresholds. The candidates are then collected by the *Muon CTP Interface* (MUCTPI), which removes the ones counted twice in overlapping regions and sends the number of candidates passing each threshold to the CTP. Similarly to the L1Calo, in case of an L1A decision, the coordinates of the muon candidates are sent to the HLT as RoIs.

The total geometrical coverage of the L1Muon reaches 99 % in the end-cap regions, while in the barrel region it is limited to 80 % due to the gaps left around $\eta = 0$ for the services of the inner detector and calorimeters, and to the support structures of the detector [48].

L1 Topological Trigger

Installed during Run 2, the Level-1 Topological trigger (L1Topo) provides another trigger input to the L1 CTP by leveraging event topology information. More specifically, it uses FPGAs to process the trigger objects coming from the L1Calo and L1Muon subsystems, together with event-level quantities such as the E_T^{miss} , computing kinematic and geometrical variables like the angular distances or the invariant masses of the objects. For Run 3, the L1Topo hardware was replaced, in order to receive the objects produced by the new L1Calo feature extractors and by the new MUCTPI [26].

This new topological information has a huge physics potential: in $H \rightarrow \tau_{had}\tau_{had}$ events, for example, requiring the two tau candidates to be close in ΔR rejects most of the background events while keeping the signal ones, allowing to lower the p_T thresholds of the tau candidates without increasing the trigger rate [49].

L1 Central Trigger

As briefly mentioned above, the role of the L1 Central Trigger is to make the final L1-Accept decision, based on the information received from the L1Calo,

L1Muon and L1Topo subsystems. Additionally, the CTP is responsible for the distribution of timing information and synchronization signals to all the detectors. By design, the trigger signal needs to be formed within 100 ns from the bunch crossing, corresponding approximately to four LHC clock cycles. Figure 3.6 shows a schematic of the L1 Central Trigger, with the different components and their connections to the other subsystems.

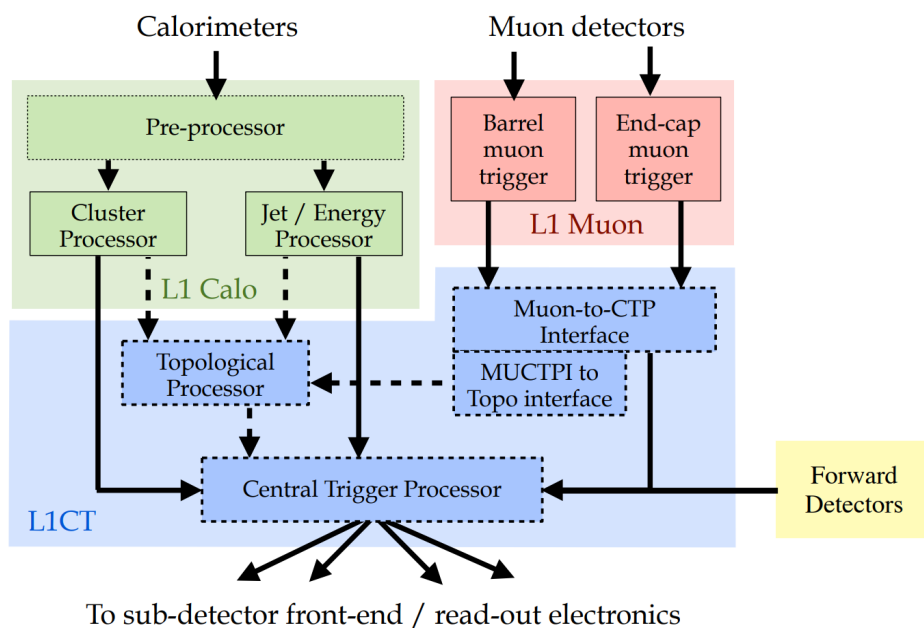


Figure 3.6: Schematic of the Level-1 Central Trigger [50].

3.3.2 Readout and Dataflow

The L1A decision taken by the CTP is distributed to all the detectors by the *Timing, Trigger and Control* (TTC) system, which also provides the LHC clock and the information needed to label each event: the *bunch crossing identifier* (BCID) and the *Level-1 identifier* (L1ID), a counter incremented at each L1A. Since all the fragments of an event carry the same identifiers, these are then used to build the event, as described in section 3.1.3.

Upon receiving an L1A, the front-end electronics move the data of the corresponding bunch crossing from the pipeline memories to the derandomizing buffers, and send them over optical links to the *Readout Drivers* (RODs), detector-specific boards that collect the data of a portion of the detector and format them into fragments. The fragments are then forwarded to the *Readout System* (ROS), made

of commodity servers equipped with custom PCIe cards, which buffers them until they are requested by the dataflow system, or deletes them if the event is rejected. The ROS has to sustain the L1A rate of 100 kHz, with a 20 % contingency [26].

For the systems installed or upgraded for Run 3, such as the NSW and the new LAr and L1Calo trigger electronics, the readout is instead based on the *Front-End Link eXchange* (FELIX) [46]. FELIX consists of FPGA-based PCIe cards installed in commodity servers, acting as a router between the custom serial links of the front-end electronics and a commodity switched network; it also forwards the TTC information to the front-end with a fixed latency. The data received by FELIX are processed by the *Software ROD* (SW ROD), running on dedicated servers; it performs the tasks of both the ROD and the ROS, building the fragments and buffering them for the dataflow system. This approach, replacing custom electronics with software running on commercial hardware, is the one adopted for the whole detector for the HL-LHC.

If the buffers of any readout component are about to be full, a BUSY signal is sent to the CTP, which stops accepting new events until the buffers are emptied: this introduces some dead time, but avoids losing data, as described in section 3.1.1.

As mentioned, the event building itself and the distribution of fragments and events are managed by the *dataflow system*, which in Run 3 follows a pull scheme, as described in section 3.1.3. Its main components are:

- **HLT Supervisor** (HLTSV): receives from the L1 system the result of each accepted event, including its identifiers and RoIs, and assigns the event to one of the nodes of the HLT farm with free processing resources. Once the processing of an event is completed, it also instructs the readout system to delete the corresponding fragments;
- **Data Collection Manager** (DCM): running on each node of the HLT farm, it receives the events assigned by the HLTSV and passes them to the *HLT processing units*, the processes running the selection algorithms. When the algorithms need data, the DCM requests the corresponding fragments from the ROS and the SW ROD, at first only the ones in the RoIs; if the event is accepted, the full event is built and sent to the output;
- **Sub-Farm Output** (SFO): receives the accepted events and writes them to files on local disks, from where they are transferred to the permanent storage at the CERN computing center. The SFO can sustain a maximum throughput of 8 GB/s, and provides a storage volume of at least 1.4 PB, enough to keep at least 24 hours of data in case the transfer to the permanent storage is

interrupted, with additional margin for the file transfer latencies and for keeping the required file system performance [26].

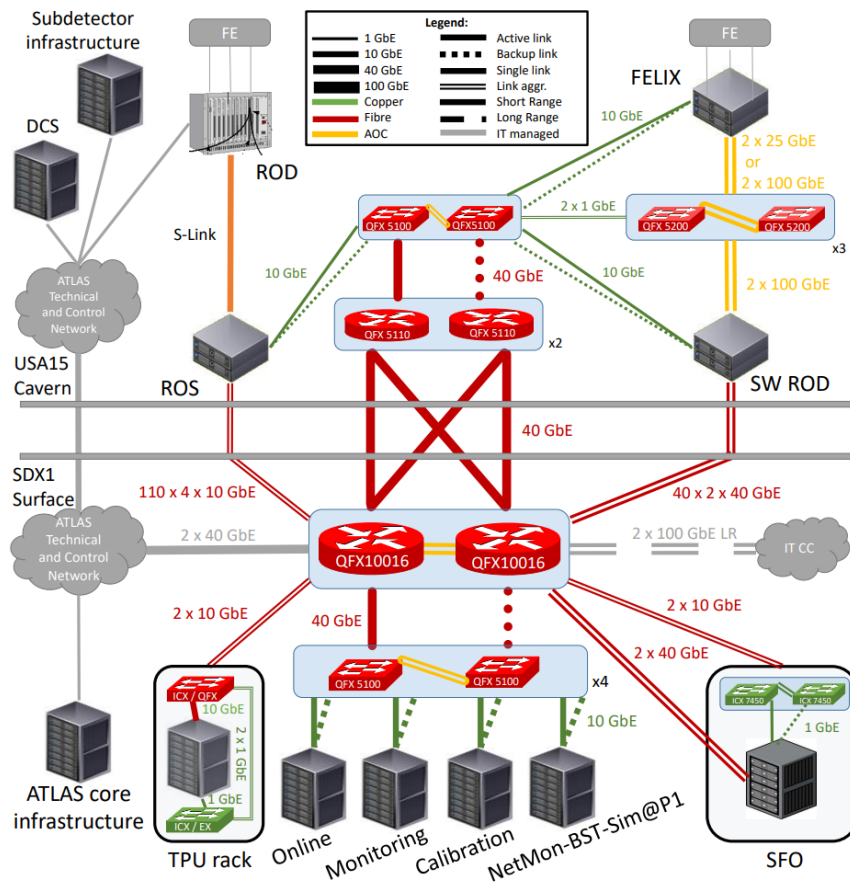


Figure 3.7: Schematic of the ATLAS TDAQ network for Run 3, showing the Run 2 ROD–ROS readout (top left) alongside the new FELIX/SW ROD system (top right). Nodes participating in primary dataflow (such as ROS, SW ROD, TPUs and SFO) are connected directly to the core [26].

The full Run 3 network structure is shown in figure 3.7. The HLT nodes, each running a DCM and several HLT applications, correspond to the racks labelled as *TPU* (Trigger Processing Unit) racks, located on the surface together with the SFO and the monitoring and control racks. In the service cavern (USA15), instead, the ROD–ROS and the FELIX–SW ROD readout systems coexist, with the former reading out the legacy detectors and the latter the upgraded ones. Much of the complexity introduced by this double readout system will be removed with the HL-LHC upgrade, when all the detectors will be read out by FELIX and, in the

current design¹, the applications replacing the SW ROD will run on the same servers hosting the FELIX cards (see section 3.4.2).

3.3.3 High-Level Trigger

As mentioned before, after the L1A decision is formed by the CTP, the information on the selected event is passed to the HLT, where more refined and complex selection algorithms are applied. These algorithms are part of *Athena* [51], the same software framework used for the offline reconstruction, following the approach described in section 3.1.2. To save processing time, the HLT usually reconstructs the event only inside the RoIs found by the L1, while for some signatures, like the hadronic ones, the tracking has to be run on the full detector.

Overall, the HLT reduces the rate from the 100 kHz of the L1 to about 3 kHz, averaged over an LHC fill [26]. In 2022, at the peak luminosity of a typical run, the processing of an event took on average about 600 ms, compared to the 500 ms of Run 2 [47]: with events arriving at up to 100 kHz, this means that about 60 000 events are being processed at the same time, requiring a farm with a comparable number of CPU cores.

It's also important to note that with the increase in luminosity and pile-up due to detector upgrades, also the bandwidth requirements of the HLT increase considerably (figure 3.8 left). Additionally, computation time does not scale linearly with pile-up conditions, as shown in figure 3.8 right: the solution can't be simply to increase the number of CPU cores, since the cost and power consumption would be prohibitive. The solution is to improve the algorithms, making them more efficient and faster, and to use hardware accelerators, such as GPUs or FPGAs, for the most computationally intensive tasks.

Trigger Menu

The trigger strategy itself is defined using the *trigger menu*, a comprehensive set of L1 and HLT trigger chains used to select specific events [47], while the specifics for the HL-LHC menu are still under study.

Triggers used can be divided into three main categories:

- **Primary triggers:** used for physics signatures, not pre-scaled².

¹This is the configuration presented in the FELIX Technical Design Report (April 2026), not yet published at the time of writing.

²prescaling is a technique used to reduce the rate of a trigger by accepting only one event out

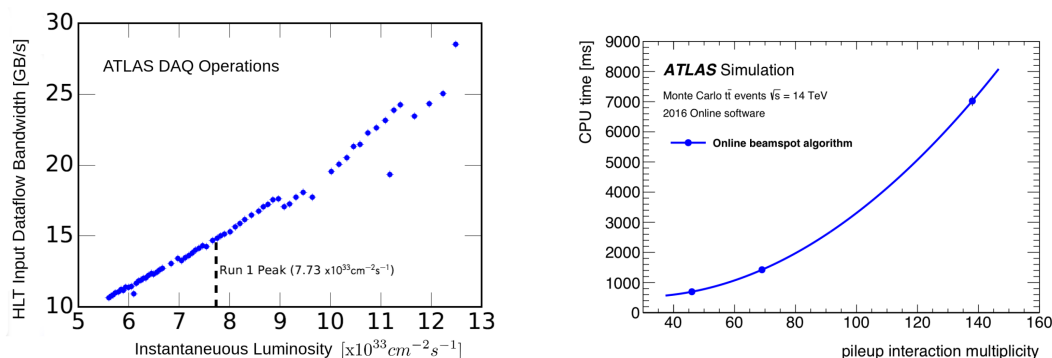


Figure 3.8: HLT bandwidth as a function of instantaneous luminosity [52] (left) and the CPU time necessary for track reconstruction as a function of pile-up in case of simulated $t\bar{t}$ at $\sqrt{s} = 14 \text{ TeV}$ [45] (right).

- **Support triggers:** used for efficiency and performance measurements, at a much smaller rate and lower threshold with respect to primary triggers.
- **Calibration and monitoring triggers:** used to collect data for detector calibration and monitoring, at a very low rate.

Moreover, triggers can operate the selection using a single property, such as the transverse momentum of a particle (*single object triggers*), or a combination of properties from the event final state (*multiple object triggers*). Also triggers taking into account the event topology are widely used (*topological triggers*), for example requiring two objects to be close in ΔR or to have an invariant mass within a certain range. The full list of triggers used during Run 3 can be found in [47].

Depending on the chains that accepted it, each event is written to one or more *streams*, recorded separately to facilitate further processing [47]. The main ones are:

- **physics streams**, containing the full detector information of the events used for physics analyses: the *Main* stream, promptly reconstructed, and the *delayed* streams, reconstructed later when computing resources allow it;
- **express stream**, a small subset of the physics events used for prompt monitoring, calibration and data quality checks;
- **calibration streams**, containing only the partial detector information needed for the calibration of specific subdetectors;

of N passing the selection, with N the prescale factor

- **debug streams**, collecting the events for which no trigger decision could be taken, for example due to crashes or timeouts in the HLT processing;
- **Trigger-Level Analysis (TLA) streams**, containing only specific physics objects reconstructed by the HLT, and optionally partial detector information.

The TLA dedicated streams, introduced in Run 3, save events to be directly used in the physics analyses, without being reconstructed offline. In 2022, their average size was 4.5 kB, about 0.3 % of a full ATLAS event: this allowed to record jet events in the TLA stream at about 5 kHz, against the about 0.5 kHz of jet triggers in the Main stream. This technique, and its application to the upgraded HL-LHC system, is the subject of chapter 5.

3.4 The Run-4 TDAQ System

To take full advantage of the physics potential of the HL-LHC, the ATLAS TDAQ system is being completely redesigned to sustain the higher event rates and sizes (table 3.1). The Level-0 trigger will operate at 1 MHz, with a latency of 10 μ s, while the HLT will reduce the rate to 10 kHz. The readout system will have to sustain a throughput of about 4.6 TB/s, more than twenty times larger than in Run 3, and the HLT farm will have to process events at a rate of 1 MHz.

Figure 3.9 shows a schematic of the Run-4 TDAQ system, with the main components and their connections. The following sections describe the Level-0 trigger, the readout and dataflow system, and the HLT with its selection menu.

3.4.1 The Level-0 Trigger

The Run-4 Level-0 trigger accepts events at up to 1 MHz with a latency of 10 μ s, as reported in table 3.1. It is composed of the calorimeter (L0Calo) and muon (L0Muon) triggers, the *Global Trigger* and the CTP, as shown in figure 3.9. L0Calo reuses the feature extractors introduced for Run 3, with the addition of a new one for the forward region (fFEX), while the longer latency allows L0Muon to also use the MDT chambers, improving the measurement of the transverse momentum [45].

The main novelty is the Global Trigger [54], which receives the full-granularity calorimeter information at 40 MHz, together with the objects identified by L0Calo and L0Muon (through the MUCTPI); it runs much more complex algorithms, closer to the offline ones, fully replacing L1Topo. Its result is sent to the CTP, which also receives the muon candidates directly from the MUCTPI and takes the final

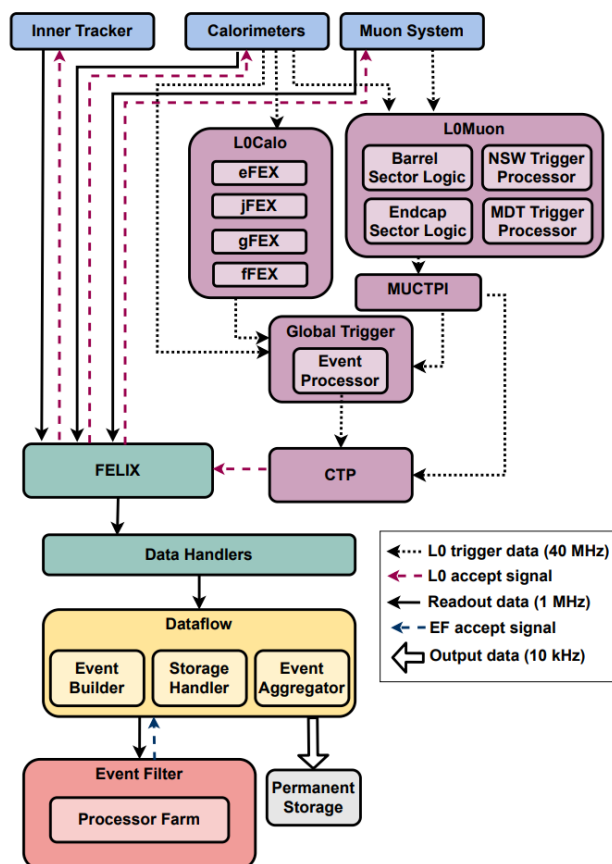


Figure 3.9: Schematic of the ATLAS TDAQ architecture for the Run-4 upgrade [53].

decision: the L0-Accept (LOA) signal is then distributed to the detectors through FELIX, together with the event identifiers.

3.4.2 Readout and Dataflow

In Run-4, the whole detector is read out through FELIX, which receives the data of the events accepted by the L0 at up to 1 MHz and, as in Run 3, distributes the L0 accept and the event identifiers to the front-end electronics. The data received by FELIX are processed by the *Data Handlers* (DH), the successors of the SW ROD, running on the same servers hosting the FELIX cards³: they collect the data coming from the front-end links, build the event fragments and keep them in memory until they are requested by the dataflow system.

³These are special 4 units servers, housing two 96 cores cpu each and 4 FELIX cards in total, two per numa node.

The dataflow system keeps a structure similar to the Run-3 one, with each Run-3 component having a functional equivalent in the Run-4 architecture (figure 3.9):

- **Event Dispatcher (ED)**: the equivalent of the HLTSV, it assigns each event accepted by the L0 to an Event Builder, and requests the corresponding fragments from the Data Handlers;
- **Event Builder (EB)**: the equivalent of the DCM, it gathers the fragments of the assigned event and passes the complete event to the Event Filter, which runs in several instances on the same node, as the HLT processing units in Run 3;
- **Event Aggregator (EA)**: the equivalent of the SFO, it receives the events accepted by the Event Filter, writes them to local disks and transfers them to the permanent storage at the CERN computing center.

The communication between the dataflow and the other systems is handled by two software interfaces: the *DataFlow Data Handler (DFDH)* interface, running on the Data Handlers, and *DataFlow Event Filter (DFEF)* interface, used by the Event Filter instances to receive the events from the Event Builder and to return their decision.

The physical layout of the system is shown in figure 3.10. The Data Handlers are going to be located underground in the USA15 cavern and connected through the core network to the surface building SDX1, where the Event Aggregators and most of the Event Filter racks are going to be installed. Each Event Filter rack is going to be connected to the network through a top-of-rack switch, while the Event Aggregators will keep sending the accepted events to the CERN storage through a dedicated link, like the SFO.

3.4.3 Event Filter

During Run 4 the HLT selection is going to be performed by the *Event Filter (EF)* software, which reduces the rate from the 1 MHz of the L0 to the 10 kHz saved to storage (table 3.1). Similarly to the HLT processing units in Run 3, several EF instances run on each Event Builder node: the EB passes them the complete events and receives back the selection decision. The accepted events are then sent to the Event Aggregators, while the rejected ones are discarded.

In the TDR, the reconstruction of the tracks in the ITk, the most demanding task for the EF, was assigned to a dedicated hardware co-processor, based on associative memory ASICs and FPGAs on custom boards [45]. With the decision not to intro-

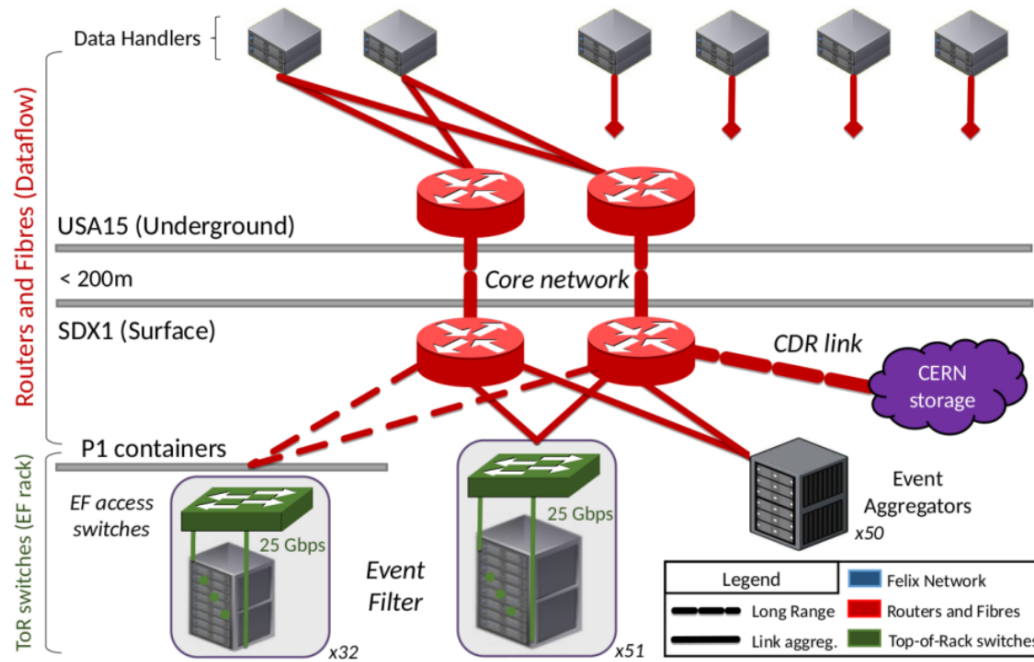


Figure 3.10: Physical layout of the Run-4 dataflow network [55].

duce a second trigger level based on tracking, the low-latency requirement behind this design was removed, and the co-processor was replaced by a heterogeneous commercial system, made of CPU cores and possibly accelerators such as GPUs or FPGAs [53]; following the technology choice made in early 2026 by the designated committee [56], in the current design the most demanding steps of the tracking will be offloaded to GPUs provided *as a service*.

Chapter 4

The HL-LHC Dataflow Upgrade

Due to the higher rates and event sizes expected at the HL-LHC (table 3.1), the ATLAS dataflow software is being redesigned as part of the Run-4 TDAQ system described in chapter 3. In this chapter, the work carried out within the ATLAS dataflow group is presented. After an introduction to the dataflow software and to the infrastructure it relies on, the interface developed to allow the new FELIX readout to operate with the Run-3 dataflow is described. The focus is then on the deployment of the Run-4 dataflow at full scale using Kubernetes, a platform for the orchestration of containerized applications, and on the monitoring of the health of its components.

4.1 The Dataflow Software

The dataflow software is part of the ATLAS TDAQ software, and consists of a set of applications running on commodity servers, exchanging data and control messages over the network. In Run 3, these are the HLTSV, the DCM and the SFO described in section 3.3.2, which have been used in production for the whole data taking and are still used to test the new detector systems before the start of Run 4. For Run 4, a new set of applications is being developed, the Event Dispatcher, the Event Builder and the Event Aggregator, whose design was presented at the Preliminary Design Review of the dataflow system [55]. Since they will only be used in production from Run 4, these applications are developed and tested on dedicated testbeds and, for the tests at full scale, on the Run-3 farm itself.

4.1.1 Software Architecture

The Run-4 dataflow software is made of three applications, the Event Dispatcher, the Event Builder and the Event Aggregator, described in section 3.4.2, and of two libraries, shown in figure 4.1:

- **DFDH**, running on the Data Handlers, which provides the fragments to the dataflow when requested;
- **DFFEF**, used by the Event Filter instances running on the Event Builder nodes to receive the complete events and to return their selection decision.

Since the Event Filter communicates with the dataflow only through DFFEF, it can be replaced by an emulator using the same library, as done in the tests of section 4.3: even if the events themselves are never processed and a timeout simulates the processing time, from the Event builder point of view nothing changes. This proved very useful during dataflow testing campaigns (described in the next sections) and allowed the development of the dataflow software to proceed in parallel with the development of the Event Filter, both still ongoing.

4.1.2 Controls and Configuration

During data taking, tens of thousands of TDAQ applications run on thousands of hosts: at this scale, hardware and software failures cannot be avoided, and the applications have to be configured, controlled and monitored in a coherent way to guarantee an efficient data taking. This is the role of the *Controls and Configuration* (CC) software, which provides both low-level services, used for example to manage the processes on each host, and high-level tools, such as the graphical interfaces used to steer the data acquisition. In Runs 1 and 2, ATLAS reached an overall data-taking efficiency of 94 %, with only a negligible fraction of the dead time due to the CC software [57].

The configuration of a data-taking session is stored in OKS (Object Kernel Support), an object-oriented configuration database [58]. Classes describe the types of components, such as hosts or applications, while objects describe the actual instances used in the session, with their parameters. The complete description of a set of applications, together with the hosts they run on, is called a *partition*: the full ATLAS data taking corresponds to one partition, but smaller partitions can be defined to run a single subdetector or a test setup independently.

The CC services communicate over the network through an inter-process communication (IPC) layer based on CORBA [59]. The main ones are:

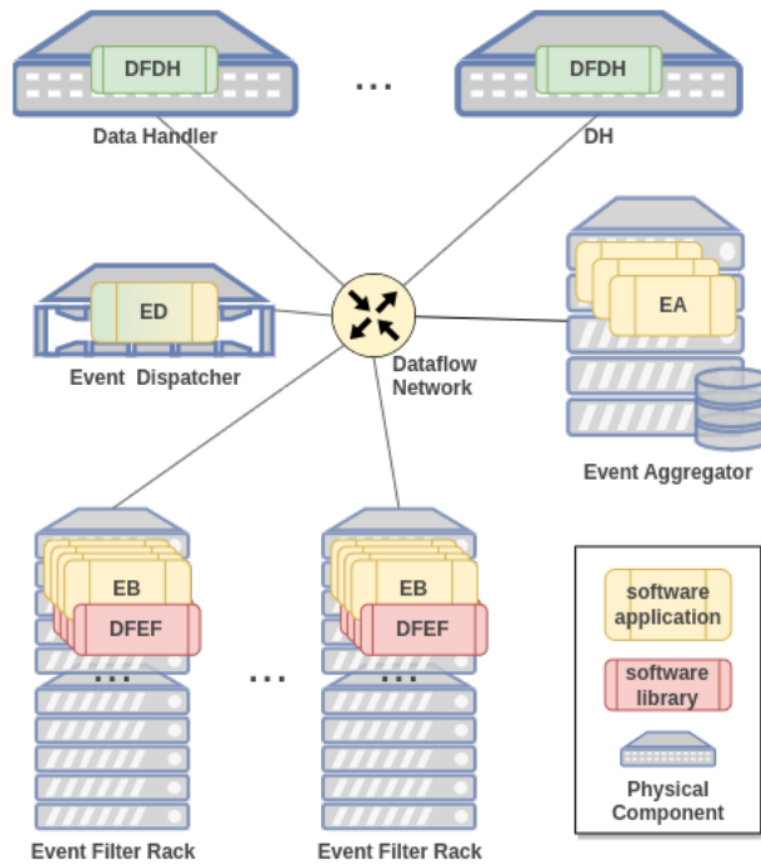


Figure 4.1: Software architecture of the Run-4 dataflow system [55]. Software components are shown overlaid on the hardware on which they will be executed. Components interacting with the Readout system are shown in green, those interacting with the Event Filter in red and the Dataflow-only ones in yellow.

- **Process Manager (PMG):** starts, stops and monitors the processes on each host;
- **Resource Manager (RM):** allocates the computing resources to the applications;
- **Error Reporting Service (ERS):** collects the errors and warnings reported by the applications, which are then stored and displayed by a dedicated log service;
- **Information Service (IS):** used by the applications to publish their opera-

tional information, such as their status, rates or buffer occupancies, read by the monitoring tools and archived for later analysis.

The information published in IS is archived by *P-BEAST* (Persistent Back-End for the ATLAS Information System), which stores all the operational monitoring data for the whole lifetime of the experiment: during Run 2, it received on average 200 000 updates per second, storing about 20 TB of compressed data per year [60]. Its data are accessed mainly through *Grafana*, a web interface used to build dashboards of time series, which allow following the evolution of any monitored quantity, such as rates, buffer occupancies or CPU usage, during and after a run [26].

The applications of a partition are coordinated by the *Run Control* [57, 61]. Its controllers are organized in a hierarchical tree, each controlling a group of applications or other controllers, and all of them follow the same finite-state machine. A command sent by the operator to the top-level controller, such as configuring the system or starting a run, is propagated down the tree, and each controller completes its transition only when all its children have done so, so that the whole partition stays synchronized. The Run Control tree of a partition, as displayed by the main graphical interface of the data acquisition, is shown in figure 4.2.

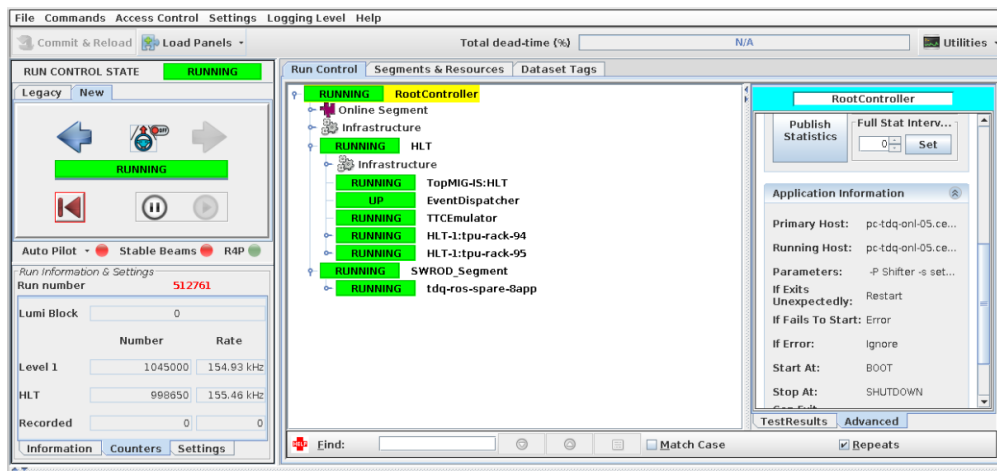


Figure 4.2: The TDAQ graphical interface (IGUI) during a dataflow test run on the Run-3 farm. The Run Control tree (center) shows the state of each controller and application, while the panel on the left reports the run information and the L1 and HLT rates read from IS.

An ATLAS *run* is a period of data taking with a stable detector configuration, identified by a unique run number and, in the case of physics data taking, usually coinciding with an LHC fill [62]. To start a run, all the applications follow the

same sequence of transitions of the finite-state machine, shown in figure 4.3:

- **boot**: all the applications are started;
- **configure** and **connect**: the hardware and the applications are configured, and the connections between the different applications are established where necessary;
- **start**: the run number is assigned and the applications perform their final, run-dependent configuration, which can take several minutes for the entire ATLAS detector.

Once all the applications have reached the ready state, the CTP releases the trigger inhibit and the events start flowing through the system.

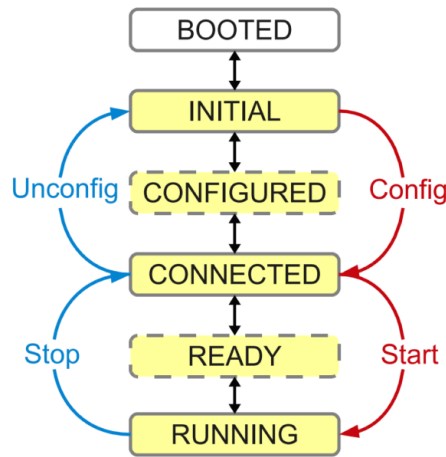


Figure 4.3: Transitions of the finite-state machine followed by the ATLAS applications to start a run [62].

Finally, to reduce the downtime caused by failures, the recovery of the system is partly automated: an expert system, the *Central Hint and Information Processor* (CHIP) [61], performs the standard recovery procedures automatically, while the *Shifter Assistant* [63] analyzes the operational parameters and suggests actions to the operators in the control room.

4.1.3 Testbeds

The development and the first tests of the software presented in this thesis were carried out on the ATLAS TDAQ testbed, a cluster at CERN dedicated to the

development and testing of the TDAQ and detector software. Its aim is to reproduce as closely as possible the online environment at Point 1: the nodes run the same operating system and software infrastructure, and the worker nodes are netbooted in the same way as the HLT nodes, reusing hardware of the same type previously installed in the HLT farm. The main part of the testbed consists of about one hundred of these nodes, each equipped with two 12-core Intel Xeon E5-2680 v3 CPUs and 64 GB of RAM, while a separate set of 32 nodes is reserved for the development of the Run-4 Controls and Configuration software on Kubernetes.

The tests at full scale were instead performed on the Run-3 HLT farm itself, the TPU racks shown in figure 3.7. In 2022, the farm consisted of about 56 000 physical cores (112 000 with hyper-threading), on which up to 90 000 HLT selection applications were running. A typical node is a dual-processor server with two AMD EPYC 7302 CPUs, each with 16 physical cores and two hyper-threads per core, and 128 GB of RAM [47]. The nodes on the surface are connected to the systems in the electronics cavern through a 40 Gbit s⁻¹ Ethernet backbone, and the traffic is separated into virtual networks dedicated to the data, the control and the monitoring [26].

In both environments, the operational monitoring data published by the applications are archived by P-BEAST and monitored through Grafana dashboards, in the same way as during data taking.

4.2 Integration of the Upgraded Run-4 Readout with the Run-3 Dataflow

The upgraded dataflow will only be used in production from Run 4, while the new FELIX hardware and software are already being tested by the subdetectors during LS3 since the FELIX project went through its *Final Design Review* (FDR) in April 2026. To perform these tests the Run-3 dataflow described in section 3.3.2 should be used: the SW ROD already receives its data from FELIX, but the HLTSV had no way to receive the identifiers of the accepted events from the upgraded Run-4 FELIX, which distributes them in a new format. For this purpose, a new input source for the HLTSV was developed as part of this thesis, the Run-4 equivalent of the TTC2LAN source (section 4.2.1) used in Run 3; with it, a subdetector can run its new FELIX readout with a standard Run-3 dataflow partition, changing only the input of the HLTSV in the configuration.

4.2.1 Input Sources of the HLTSV

The HLTSV receives the accepted events through an *input source*, following the *plugin* design pattern, in which the implementation to be used is linked during the configuration rather than at compilation [64]. All the sources implement the same interface¹ and are compiled in separate shared libraries: the OKS configuration of the HLTSV application specifies the library to use, which is loaded during the configure transition, and a factory function with a fixed name creates the source. The rest of the HLTSV interacts with the source only through the common interface, thus not depending on the specific source being used. For each event, the source provides the L1ID and the L1 result, which the HLTSV sends to the DCM together with the event assignment. The L1 result is made of one or more *ROB fragments*, the same format used for the detector data, named after the *Read-Out Buffers* (ROBs) in which the ROS stores them [47]. The sources available in Run 3 were:

- **RobinNP**: used during data taking, it receives the L1 result from the *Region of Interest Builder* (RoIB), which assembles the fragments sent by the L1 systems into a single record; since Run 2, the RoIB is implemented in software on a RobinNP card, in the same server running the HLTSV [66];
- **Internal**: generates the events and their CTP fragment internally;
- **Preloaded**: reads the L1 results from a raw data file;
- **TTC2LAN**: receives the L1IDs over the network from a separate application.

TTC2LAN is mainly used to receive the event identifiers from the local trigger interface of a subdetector, for example during calibration runs, while in the dataflow test partitions they are produced by a software generator. The identifiers are sent to the HLTSV over a TCP connection, whose endpoint is published through the IPC name service. Each message only contains the 32-bit L1ID of the last accepted event, and the source generates all the identifiers between the previous one and the received one.

In Run 3, this *extended L1ID* is made of a 24-bit counter, incremented at each L1A, and of an 8-bit counter of the *Event Counter Resets* (ECR), the signal that resets the first one. When the upper 8 bits change, the source interprets it as an ECR and restarts the sequence from the new value. To regulate the event flow, if the number of events waiting in the HLTSV exceeds a configurable threshold, the

¹The use of a common interface with interchangeable implementations corresponds to the *Strategy* pattern, whose intent is to “define a family of algorithms, encapsulate each one, and make them interchangeable” [65].

source asks the sender to stop, and lets it resume once they drop below a lower one (XON/XOFF mechanism).

Since TTC2LAN provides only the event identifiers, without any information from the CTP or from the other L1 systems, the source has to build the L1 result itself. The DCM and the HLT cannot process an event without a CTP fragment, so a *dummy* CTP fragment is created for each event. Its payload is a fixed template, taken from a real CTP fragment of a raw data file, in which only the timestamp is updated, while the header contains the run number and the L1ID of the event, with the BCID and the trigger type set to fixed values. The resulting fragment is correctly formatted, ensuring the proper functioning of the dataflow infrastructure, but the content itself has no meaning for the rest of the system. The data taken with TTC2LAN, typically during calibration runs, are processed, if at all, by dedicated tasks that only read the subdetector fragments.

4.2.2 TTC Information in Run-4

In Run-4, the TTC information is distributed to FELIX by the *Local Trigger Interface* (LTI) modules. Upon each L0-Accept, the FELIX firmware generates a message on a dedicated *TTC ToHost* virtual e-link², which can be subscribed to like any other e-link, so that the event data can be matched to the TTC information [67]. Then, on the FELIX server, the `felix-tohost` application publishes each e-link on the network under a unique 64-bit *FELIX ID* (FID), which any other application can use to subscribe to the TTC e-link, as the SW ROD does and the Event Dispatcher will do in Run-4.

Figure 4.4 shows the message format, which has a length of 20, 22, 24 or 32 bytes, depending on the optional fields included. Besides the BCID and the trigger type, it contains the identifier of the event, the L0ID, the luminosity block (LBID) and the LHC orbit counter (OrbitID). In the format used before firmware version 5.3, the message still contained a 32-bit extended L1ID with the same structure as in Run 3, which in the current format is replaced by the 38-bit L0ID, without an ECR counter. This is one of the main incompatibilities with the Run-3 dataflow, where the event identifier is a 32-bit field both in the HLTSV and in the event format.

²The data transmitted over each optical link of FELIX are organized in configurable logical streams, called *e-links*, with variable bandwidths depending on the needs of the front-end electronics [46].

Word#	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0	ErrorFlags(3:0)				BCID(11:0)												Length 20, 22, 24 or 32 (7:0)								FMT (4) (7:0)											
1	TriggerType(15:0)																Reserved (8:0)								PT		Part(1:0)		ltx_trig (3:0)							
2																	LOID(31:0)																			
3	LBID (15:0)																ltx_tag(7:0)								Rsvd(1:0)		LOID(37:32)									
4																	OrbitID(31:0)																			
5	SyncUserData (15:0) (optional)																SyncGlobalData (15:0) (optional)																			
6																	AsyncUserData(31:0) (optional)																			
7																	AsyncUserData(63:32) (optional)																			

Figure 4.4: Format of the message sent by FELIX on the TTC ToHost virtual e-link upon each L0A, for firmware version 5.3 and later [67].

4.2.3 The FELIX Input Source

The new source receives the TTC messages by subscribing to the TTC e-link through the FELIX client library. In the OKS configuration, it only requires the FID of the TTC e-link, the network interface and the FELIX bus to use for the subscription, and the size of its internal buffer; the rest of the partition is identical to a Run-3 one, as shown in figure 4.5. The source is included in the TDAQ software from release 14.

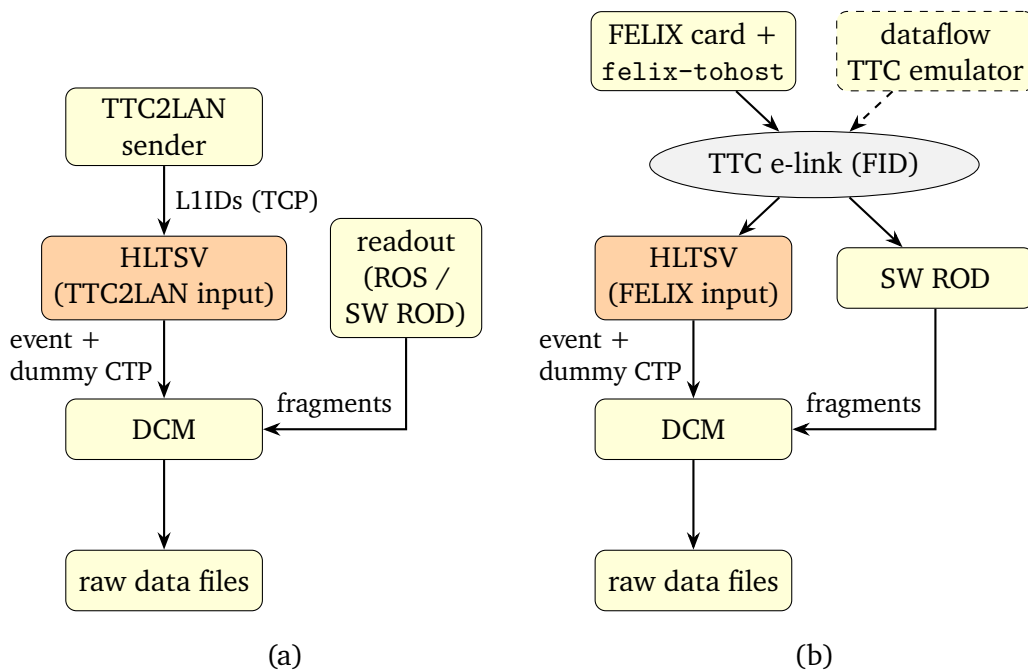


Figure 4.5: Run-3 dataflow test partition with the TTC2LAN input source (a) and with the FELIX input source (b). Only the input of the HLTSV (in orange) changes between the two. In (b), the TTC messages can be published either by `felix-tohost` from a FELIX card or by the dataflow TTC emulator (dashed), and the SW ROD receives the detector data from FELIX through the data e-links (not shown).

First, the subscription is attempted during the configure transition; since `felix-tohost` may not be publishing yet at that moment, a separate thread retries it every second until it succeeds. Each message received is then processed as follows:

- the L0ID is extracted from the message, both from the full format of figure 4.4 and from the shorter one produced by the dataflow TTC emulator (section 4.2.4), which contains the first identifier and the number of consecutive events;
- if the L0ID does not fit in 32 bits, an error is reported and the identifier is truncated;
- the identifier is stored in a lock-free queue, from which the HLTSV takes the next event to assign.

If the queue is full, the source stops reading new messages until the HLTSV frees some space, and the backpressure is propagated to FELIX, replacing the XON/XOFF mechanism of TTC2LAN.

Then, for each event the CTP fragment is built exactly as in the TTC2LAN source, using the same template and fixed values. This was done on purpose, since the source is meant as a compatibility layer between the upgraded FELIX readout and the Run-3 dataflow to allow for tests until the upgraded dataflow system is ready to be distributed.

The main functional limitation of the new source comes from the small identifier size, compatible with the Run-3 standard (32 bits). This implies that only L0IDs below 2^{32} can be used: at the nominal L0 rate of 1 MHz this range would be covered in about 72 minutes, enough to serve the original purpose of short validation tests.

4.2.4 Validation and Use

The source was first tested on the TDAQ testbed (section 4.1.3), in two configurations:

- **without FELIX hardware:** the TTC messages are generated by the TTC emulator developed by the dataflow group for testing, which publishes them on the FELIX bus in the same way as `felix-tohost`;
- **with a FELIX card:** the L0As are generated by the card using its internal TTC emulator (`fttcemu`) and published by `felix-tohost` as in a real readout setup.

The second configuration makes the dataflow software transparent to readout and subdetector changes, since only a TTC e-link is expected to exist and can be swapped with another setup.

A sample partition for both configurations was prepared for the subdetector groups, together with the relevant documentation needed to adapt it to their setups.

The source was then used by the *Liquid-Argon Calorimeter* (LAr) group in its Run-4 FELIX test setup, with release 14 of the TDAQ software. Since no LTI was yet integrated with this release, the LOAs were generated on the FELIX card at 1 kHz, from its local clock, and distributed through the chain shown in figure 4.5. The SW ROD and the HLTSV both subscribed to the TTC e-link and the events built by the DCM were written to raw data files. During the tests:

- all the events contained the fragment of the SW ROD and the dummy CTP fragment;
- L1IDs were consecutive, without duplicates or gaps, consistent between the event header and the two fragments.

The LAr group was thus able to connect its Run-4 FELIX setup to a complete readout chain and to record events, as intended for this subdetector tests.

4.3 Deployment of the Event Builder on Kubernetes

In Run-4, the Event Filter farm will be orchestrated by Kubernetes, with the applications running on it deployed as software containers. Their deployment, the restart of the failed instances and the scaling of the processing capacity with the data-taking conditions are then handled automatically, improving the fault tolerance and simplifying the management of the farm [68]. The same approach also allows the sharing of the farm with offline production, to which it is expected to provide on average 42 % of its computing capacity; in fact, with container orchestration, the simulation could run in parallel with data taking whenever the resources are not needed by the Event Filter [53]. Only the two applications running on the Event Filter nodes, the Event Builder and the Event Filter, are going to be deployed in this way. All the other TDAQ applications, such as the Event Dispatcher and the Data Handlers, will remain under the control of the Run Control. The deployment of the Event Builder on Kubernetes was developed as part of this thesis, both on the TDAQ testbed and at full scale on the Run-3 HLT farm.

4.3.1 Containers and Kubernetes

A *container* packages an application together with its runtime dependencies, so that it behaves in the same way on any host [69]. Similarly to a virtual machine, a container has its own filesystem, share of CPU and memory, and process space. But, differently from a full virtual machine, a container doesn't run a complete operating system on top of virtualized hardware but shares the host kernel, making it much lighter (figure 4.6). The container isolation is provided by two features of the Linux kernel:

- *namespaces* give the processes inside a container their own isolated instance of a global resource, for example the process IDs, the network devices and ports, or the inter-process communication resources [70];
- *control groups* (cgroups) limit and monitor the CPU, memory and other resources used by a group of processes [71].

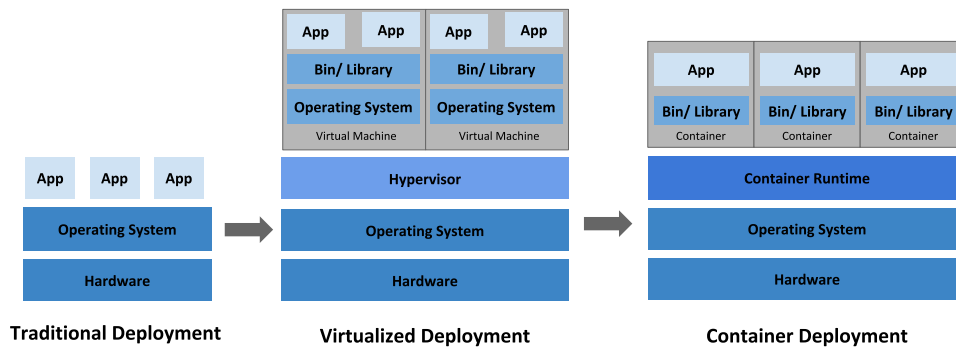


Figure 4.6: Evolution of application deployment: applications running directly on a physical server (left), in virtual machines with their own operating system (center), and in containers sharing the operating system of the host (right) [69].

The software is distributed as a *container image*, a ready-to-run package with code, libraries and default settings needed by the application to be run. Since images are immutable, changing the software would mean building a new image and recreating the containers from it. To ease this process, built images are stored in *registries* from which the hosts can download them when a container has to be started (they can be configured to re-download the image only when a new version is available).

When the farm scales up and thousands of containers have to be run on thousands of hosts, their placement, startup and recovery are managed by an *orchestrator*. Kubernetes is an open-source platform for managing containerized workloads

and services through declarative configuration and automation, and it is the de-facto standard for container orchestration, from small clusters to full-fledged data centers. It was largely derived from the experience gained at Google with the cluster managers Borg and Omega [72].

A Kubernetes *cluster* is made of a *control plane* and a set of worker machines (*nodes*) on which the containers run (figure 4.7). The main components of the control plane are:

- **kube-apiserver**: exposes the Kubernetes API and it's used by users and by all other components to access the cluster;
- **etcd**: a consistent and highly available key-value store holding all the cluster data;
- **kube-scheduler**: assigns each new pod (a group of containers, described below) to a node, according to its resource requirements and placement constraints;
- **kube-controller-manager**: runs the controllers described below.

On top of that, an agent called *kubelet* runs on each node making sure that the assigned containers are running, interacting with the container runtime that executes them.

Each Kubernetes object describes the *desired state* of a part of the system, for example the number of copies of an application that should be running. The objects can be managed in three ways:

- **imperative commands**: the user operates directly on the live objects passing the operation and its parameters to the `kubectl` command-line tool;
- **imperative object configuration**: the user still specifies the operation, but the objects are fully defined in files (usually YAMLS³);
- **declarative object configuration**: the user only provides the files, then the operations needed to create, update or delete each object are detected automatically.

In all cases, the desired state is then reached by the *controllers*, control loops that watch the cluster and make or request changes to bring its current state

³YAML (*YAML Ain't Markup Language*) is a human-readable text format for structured data, where information is written as key-value pairs and the hierarchy is expressed by indentation.

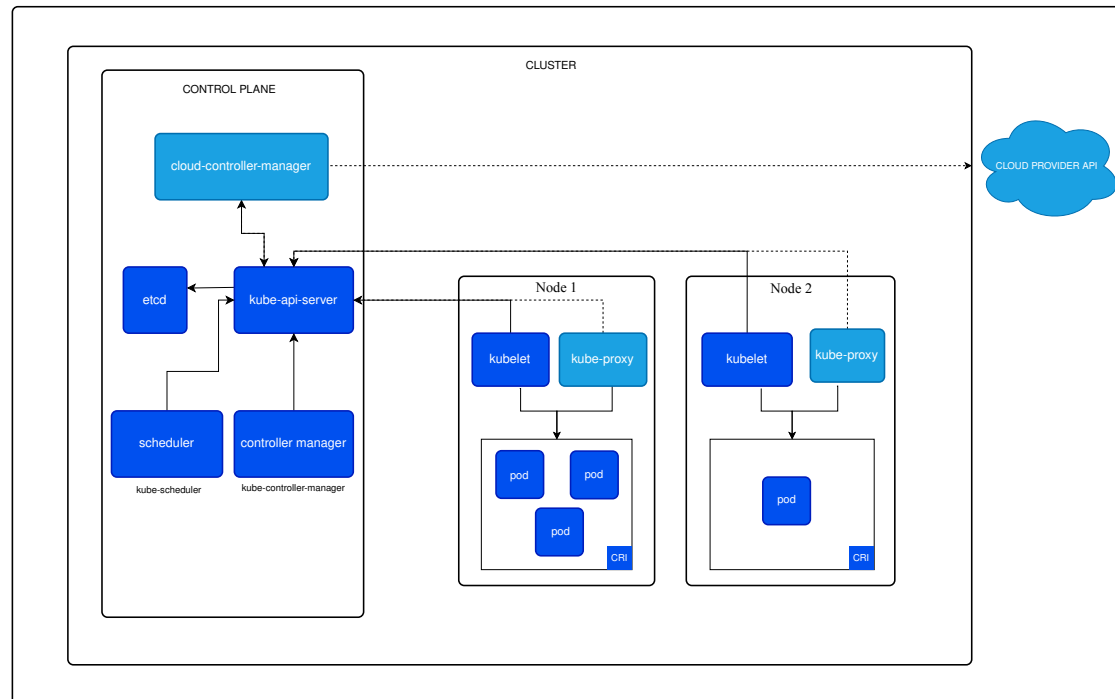


Figure 4.7: Components of a Kubernetes cluster. The control plane (kube-apiserver, etcd, kube-controller-manager, kube-scheduler) and several nodes, each running a kubelet and kube-proxy [69].

closer to the desired one. Controllers thus also handle failures automatically, since they recreate any container that terminates or is lost when the entire node it was running on fails.

The smallest unit managed by Kubernetes is the *pod*, a group of one or more containers always scheduled on the same node and sharing storage and network resources. Besides the application containers a pod can contain other container types:

- *init containers*, which run to completion before the others can be started (determining the *initialization* of the pod);
- *sidecar containers*, which run alongside the main application containers, providing supporting features such as logging or monitoring.

Moreover, pods are rarely created directly, but rather through higher-level workload objects whose controllers create and manage them:

- **Deployment:** manages a set of identical pods, keeping their number equal

to the requested number of *replicas*;

- **DaemonSet:** ensures that all the nodes (or a selected subset of them) run one copy of a pod, adding it automatically when a node joins the cluster.

The node on which each pod runs is chosen by the scheduler. Each container declares the CPU and memory it needs as *requests* and the scheduler only places the pod on a node with enough free resources; it's possible to set also a memory *limit* above which the container is terminated, to avoid excessive memory consumption that could starve other containers. Moreover, pods can be restricted to a given set of nodes using *node affinity* rules or spread evenly across nodes with *topology spread constraints*.

Concerning isolation, pods and containers are extremely similar: by default they have their own network namespace, with an IP address on a virtual network internal to the cluster, as well as their own inter-process communication namespace. Data outside container images are only accessible through *volumes*, such as host directories or network filesystems. As described in section 4.3.2, part of this isolation has to be removed for dataflow applications to guarantee proper interactions with Run Control management.

Finally, Kubernetes is highly extensible: new types of objects can be easily added to the Kubernetes API defining a *Custom Resource Definition* (CRD). The new objects (*custom resources*) are usually managed by their own controller. This corresponds to the *operator* pattern, made to allow the procedures needed to run a specific application to be written in software [69]. The KEFController described in section 4.3.3 follows this approach.

4.3.2 The Containerized Event Builder

The Event Builder is built around a core, which handles the flow of events, and four modules, each dealing with a different component of the dataflow system (figure 4.8). First, the *LO Source* receives the events assigned by the Event Dispatcher and the *Event Collector* gathers their fragments from the Data Handlers; then, complete events are handed to the Event Filter by the *Event Processor*, which also collects the selection decisions, and finally the *Event Storage* sends the accepted ones to the Event Aggregator. Every module can be replaced by an emulator, running either as a separate application or inside the Event Builder itself: the Event Builder can therefore be tested alone, or together with any part of the real system, keeping in mind that from the Event Builder point of view there's no difference between a real module and its external emulator, since they communicate through

the same interfaces.

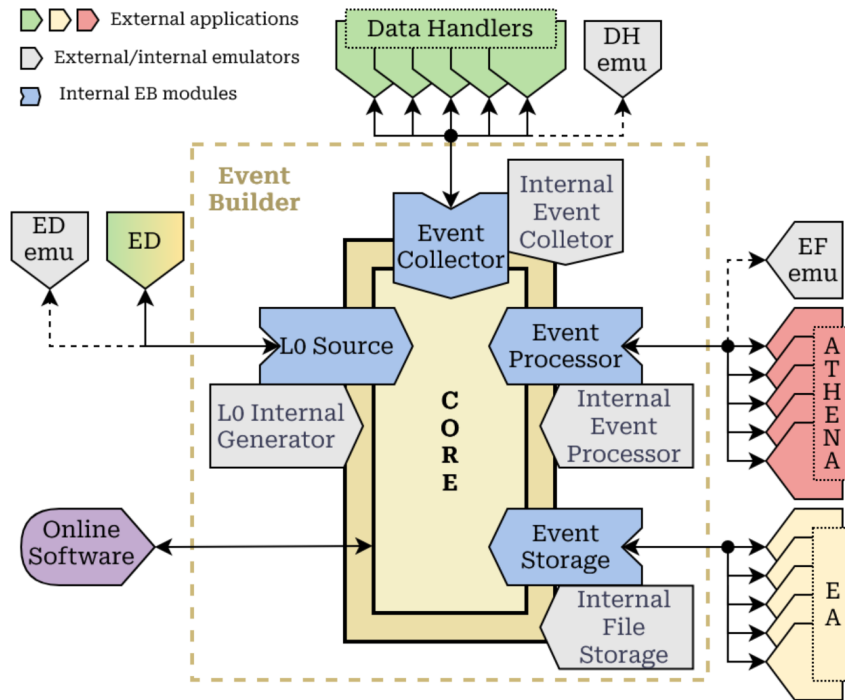


Figure 4.8: Design of the Event Builder: the core and its four modules, each with the corresponding external or internal emulator.

The Event Builder and the Event Filter instances running on the same node communicate through DFEF, using both a TCP socket and shared memory. This has two consequences on the configuration of the pods:

- they use the inter-process communication namespace of the host (hostIPC), so that the shared memory created by the Event Builder is directly visible to the Event Filter;
- the Event Filter cannot start before the Event Builder is ready to accept connections; when ready, the Event Builder writes the port of its server to a file in the /tmp directory of the host, mounted in all the pods, and each Event Filter pod contains an init container waiting for this file to appear and for the port to be open.

Host networking (hostNetwork) is used as well, since the network is already managed by the TDAQ infrastructure and the virtual network of the cluster would only add overhead (and could also generate unexpected conflicts, since the Run

The Event Builder is deployed as a DaemonSet, following the design constraint of running only one instance per node; the Event Filter instances use instead a Deployment, whose replicas are spread evenly across the nodes with topology spread constraints. On top of that, shared node affinity rules restrict both deployments to a given set of nodes. The final deployment layout is still under discussion, as several Event Filter containers could also be grouped in a single pod, or even all the applications of a node; dataflow will have to support any of these options. A scheme of the current deployment layout is shown in figure 4.9.

Figure 4.9: Layout of the pods on an Event Filter node. The Event Builder (DaemonSet) and the Event Filter emulators (Deployment) run in separate pods, sharing the host network, the inter-process communication namespace and the /tmp directory. The init container of each Event Filter pod waits for the port file written by the Event Builder before the emulator is started.

4.3.3 Deployment Models

The containerized applications can be deployed on the cluster in two different ways, both used in this work:

- *manual* deployment: the most direct, Kubernetes objects are written by hand and applied to the cluster using the `kubectl` utility, completely bypassing the Run Control;
- using the *KEFController* application: it is a Run Control application developed by the Controls and Configuration group to deploy the Event Filter, which in the production model will also be used for the Event Builder.

Both methods fall into the declarative object configuration category of section 4.3.1.

In the manual approach, each object is defined in its own YAML file, numbered according to the order in which it has to be created, since `kubectl` follows alphabetical ordering. Both the namespace and the volumes definitions have to be applied every time and they are applied first, followed by Event Builder DaemonSet and the Event Filter Deployment definitions. Since Kubernetes knows nothing about the partition being run, when applying the object configuration the synchronization is entirely up to the operator, who has to ensure that the rest of the partition (in particular the IS and RDB servers) is up and configured. Manual deployment is therefore simple and extremely flexible, ideal for development, debugging and standalone tests, but clearly not suited for data taking.

The *KEFController* (*Kubernetes Event Filter Controller*) fills exactly this gap, acting as a bridge between the Run Control and Kubernetes [68]. Being a Run Control application and thus part of the Run Control tree, it receives the same commands as any other controller and has to implement Run Control state transitions. The Kubernetes application deployment or removal can then be configured to happen at the needed FSM transitions, improving the compatibility between Kubernetes managed and Run Control managed components.

The applications are described by a custom resource called `RunControl`, defined through a CRD. Its specification wraps a standard Deployment or DaemonSet, adding a few options handled directly by the controller, like host networking, topology spread constraints, the mounting of CVMFS⁴ and of network filesystems, and a number of replicas defined per container (allowing several identical containers to run in the same pod). Following the operator pattern, the *KEFController*

⁴The *CernVM File System* (CVMFS) is a read-only, network-based file system used to distribute the software releases of the LHC experiments.

then compares the status of these resources with the desired one and applies the changes needed, by default every 3 s.

In the test partition developed in the context of this thesis work, the KEFController has its own segment, together with the list of applications to deploy (figure 4.10). For each of them, the configuration specifies the path of the YAML file and the transitions at which the application is deployed and removed as well as those at which the controller checks that these operations were completed. Deployment can happen at any point between the startup of the KEFController application and the start of a run, while removal can be triggered when the HLT is stopped or at the disconnect or unconfigure transitions (figure 4.3). The namespace, named after the partition, is created automatically, and the cleanup of all the objects requires no intervention from the operator.

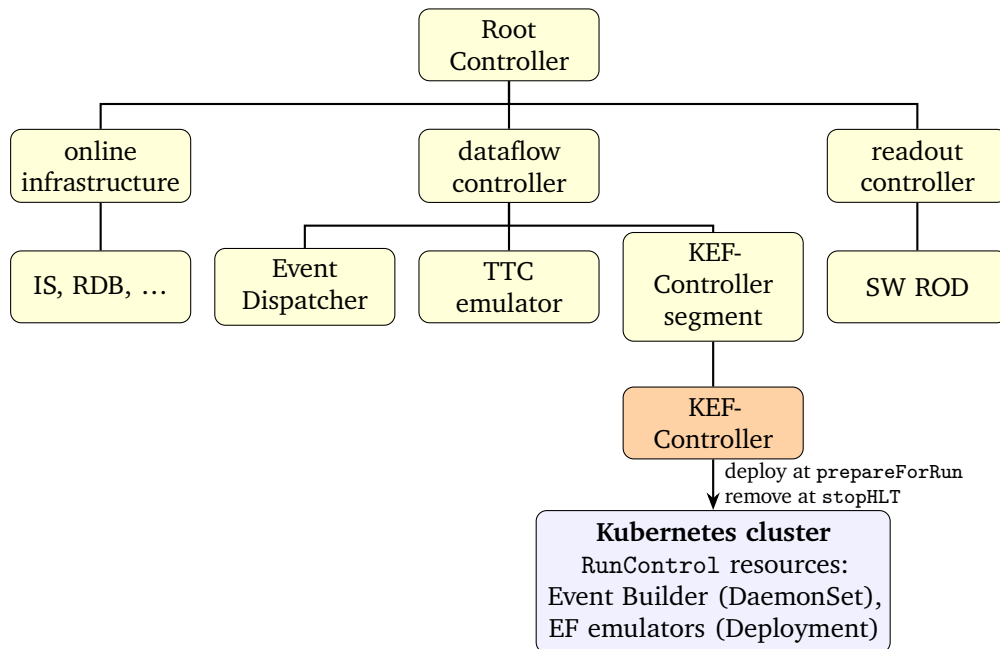


Figure 4.10: Run Control tree of the test partition using the KEFController. The KEFController runs as a Run Control application in a dedicated dataflow segment, and deploys the Event Builder and the Event Filter emulators on the Kubernetes cluster, described as RunControl custom resources, at the configured transitions.

For the Event Builder, the manual YAML files used for initial testing were translated into RunControl resources, configuring the partition to deploy both the Event Builder and the Event Filter emulators at the start of each run and to remove them at the HLT stop. Backporting the KEFController to the TDAQ release in use at Point 1 required too much effort [73]: it was therefore tested only on the testbed,

while the tests at Point 1 described in section 4.3.4 used manual deployment.

4.3.4 Deployment Tests

The Event Builder deployment was first developed and tested on the TDAQ testbed, then tested at full scale at Point 1 at the end of Run 3. The Point 1 tests were part of a larger test campaign for readout upgrades, in which the New Small Wheel was read out through a Run-4 FELIX card and a Data Handler running on a Run-4 candidate server, with the Event Dispatcher (running on a spare SWROD machine) receiving the TTC information from FELIX [73]. The Event Builders ran instead on the HLT farm, where 72 racks were switched to Kubernetes mode by sysadmins, while one was left in standard TDAQ mode to run some debug Event Builder instances. Since the KEFController was not available in the TDAQ release installed at Point 1, all the pods were deployed manually.

During the Kubernetes tests the L0 Source and the Event Collector were the real modules, receiving events from the Event Dispatcher and fragments from the Data Handlers. The Event Filter was instead replaced by the emulator of section 4.1.1, connected through the real Event Processor, while no Event Storage was used since the Event Aggregator module has not been fully developed yet. The processing time of the emulator was deliberately kept low (1 ms per event [73]) in order to avoid interfering with the main test goal of stressing dataflow components and their deployment; reproducing the long and irregular processing times of Athena would have simply moved the bottleneck to the Event Filter.

About the deployment strategy, one pod per application was chosen as the most demanding case for the deployment procedure. Similarly, the number of Event Filter instances was set to three per node, resulting in more than 8000 pods in total: those were enough to reach 1 MHz and to stress the infrastructure, without stalling the tests as tens of thousands of pods would have done.

The deployment procedure itself proved straightforward apart from a slight delay to allow the container image to be downloaded on each node, a process that took almost one hour. This was mostly due to the local registry of Point 1, never optimized for mass deployments since they were never needed before, and to the 1 Gbit s^{-1} bandwidth of the control network; for the following runs, dummy pods were spawned to keep the image cached on each node. All the 8972 pods of the full-scale test were then created in about 12 minutes (figure 4.11).

This concentrated startup, however, caused several transient errors in the TDAQ infrastructure, as all the Event Builders connected at the same time to the IS and

Chapter 4. The HL-LHC Dataflow Upgrade

```
$ kubectl get daemonsets,deployments -n rmastrof-test
```

NAME	DESIRED	CURRENT	READY	UP-TO-DATE	AVAILABLE	NODE SELECTOR	AGE
daemonset.apps/eb	2252	2252	2252	2252	2252	<none>	29m

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/ef-emu	6720/6720	6720	6720	29m

Figure 4.11: Status of the Event Builder DaemonSet and of the Event Filter emulator Deployment in the full-scale test at Point 1: all the 2252 Event Builder pods and the 6720 Event Filter emulator pods are ready.

RDB servers of the partition (figure 4.12). Some IS servers died and were restarted, and the RDB servers reported warnings, but all these issues cleared up as soon as the pods completed their initialization.

TIME	SEVERITY	APPLICATION	NAME	MESSAGE
18:19:58	WARNING	RDB_POOL_1	rdb::CannotUnsubscribeServer	cannot remove 'RDB_POOL_1' server subscription from master server 'RDB' in partition 'dataflow_p1'
18:19:24	WARNING	RootController	rc::ApplicationExited	Application "DFConfig" on host "pc-tdq-onl-27.cern.ch" exited with exit code "3". Logs are "/logs/tdaq-1.2-00-00/dataflow_p1/DFConfig_pc-tdq-onl-27.cern.ch_1782490458.out/err".
18:19:23	FATAL	DFConfig	is::ServerAlreadyExist	IS server 'DFConfig' already exists in the 'dataflow_p1' partition
18:19:22	WARNING	RootController	rc::ApplicationGenericIssue	Issue with application "mts-worker-2": the application could not be started within the init timeout
18:18:46	WARNING	EventDispatcher	eventdispatcher::EventTimeout	10 events timed out, event builder: 10.148.195.45:59812[EventBuilder]
18:18:13	ERROR	CHIP	chip::msg::TestFailed	Test failure for application: DDC , global result: FAILED Component: pc-tdq-onl-08.cern.ch , result: FAILED Follow-up action: reboot parameters: { Host: pc-tdq-onl-08.cern.ch } diagnosis: The host is in such a bad shape that it has to be rebooted.
18:18:07	WARNING	EventDispatcher	eventdispatcher::EventTimeout	10 events timed out, event builder: 10.148.203.50:41714[EventBuilder]
18:17:40	ERROR	CHIP	chip::msg::TestFailed	Test failure for application: opmon_next , global result: FAILED Component: pc-tdq-mon-03.cern.ch , result: FAILED Follow-up action: reboot parameters: { Host: pc-tdq-mon-03.cern.ch } diagnosis: The host is in such a bad shape that it has to be rebooted.
18:17:40	ERROR	CHIP	chip::msg::TestFailed	Test failure for application: monitoring-conductor , global result: FAILED Component: pc-tdq-mon-03.cern.ch , result: FAILED Follow-up action: reboot parameters: { Host: pc-tdq-mon-03.cern.ch } diagnosis: The host is in such a bad shape that it has to be rebooted.
18:17:40	ERROR	CHIP	chip::msg::TestFailed	Test failure for application: Monitoring , global result: FAILED Component: pc-tdq-mon-03.cern.ch , result: FAILED Follow-up action: reboot parameters: { Host: pc-tdq-mon-03.cern.ch } diagnosis: The host is in such a bad shape that it has to be rebooted.

Figure 4.12: Transient errors of the TDAQ infrastructure during the startup of the pods in the full-scale test at Point 1 [73].

With the whole farm deployed, enough processing slots were available to sustain an L0 rate of 1 MHz, the Run 4 target. Figure 4.13 shows the long run performed at this rate: as the rate was increased the number of available slots decreased, never reaching zero; the L0IDs inserted by the Data Handler and requested by the Event Builders stayed perfectly synchronized. Each request was sent with a delay of 20 ms, so that the fragments were already built when requested and the average collection time stayed close to this minimum, apart from a spike caused by a problematic ROB. The number of available slots was nevertheless lower than expected, given the short processing time of the emulator, and is still under investigation.

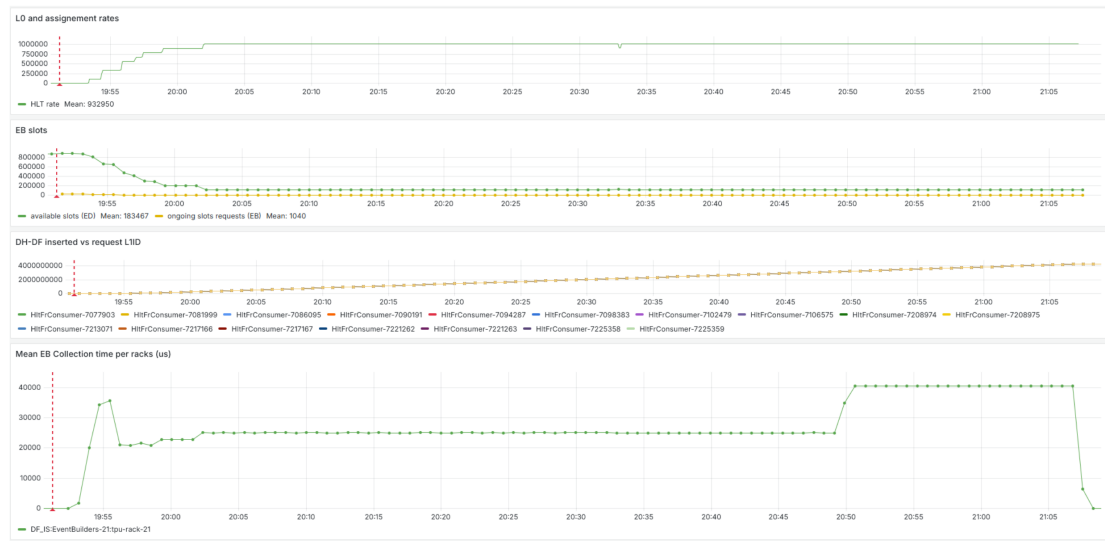


Figure 4.13: Long run at 1 MHz at Point 1. From top to bottom: L0 rate, available Event Builder slots, inserted and requested L0IDs, average fragment collection time. The collection time is always bigger than 20 ms due to a fixed delay introduced in the DH-DF interface for test purposes. Its increase to about 40 ms around 20:50 was caused by a single ROB receiving corrupted TTC messages, which caused the saturation of its fragment-building CPU thread [73].

Finally, adding racks to the running system worked without issues, showing that the capacity of the farm can be increased dynamically. Removing them, on the other hand, caused the Event Dispatcher to crash due to the fact that unexpected Event Builder disconnections are not yet handled correctly. For the same reason, when about 30 Event Builders crashed simultaneously after roughly one hour of continuous operation, the failure propagated to the Event Dispatcher and stopped the run. These issues, together with those of the infrastructure, are discussed in the next section.

Open Issues

Most of the open points concern the integration of the containerized applications with the rest of the TDAQ infrastructure. The most evident one is the observed startup surge: when thousands of pods are created at the same time, all the Event Builders contact the IS and RDB servers of the partition within a few minutes, overloading them. In the current infrastructure, the RDB servers are organized in a hierarchy following the segments of the partition, meaning that they don't automatically redistribute requests if a server gets overloaded. The issue has been reported to the Controls and Configuration group, and a possible solution, still to

be developed, could be to replace or complement the hierarchy structure with a pool of RDB servers to distribute requests to.

Moreover, the containerized Event Builder does not follow the state transitions of the partition. With manual deployment, it simply starts when its pod is created, connecting to IS and RDB through the environment variables described in section 4.3.2. The KEFController creates and removes the pods at the configured transitions, but the application itself still receives no Run Control command, and how the Event Builder should be integrated with the finite-state machine is still an open question.

On the dataflow side, the Event Dispatcher has to handle the unexpected Event Builder disconnection, which currently causes the Event Dispatcher to crash, as observed both when removing racks and when about 30 Event Builders crashed simultaneously. The cause of these crashes (that produced no core dumps) is still unknown, as well as the reason for the lower than expected number of available slots. Since both issues are independent of the detector data, they can be investigated directly on the testbed.

4.3.5 Health Monitoring of the Event Filter

Kubernetes provides two levels of protection against failures of the containers. The first one is the restart policy of the pod: by default, a container that terminates for any reason is automatically restarted. The second one is based on *probes*, diagnostics performed periodically by the kubelet on each container, either executing a command inside it or sending it a network request (HTTP, TCP or gRPC). Three kinds of probes are available [69]:

- *Startup* probes check whether the application has completed its initialization and postpone the other two until it has.
- *Readiness* probes determine when a container is ready to accept traffic; until then Kubernetes does not distribute requests to the not-ready pod.
- *Liveness* probes determine when a container has to be restarted, for example in case of deadlocks or timeouts (the application stuck for too long).

Each probe is configured with the interval between two checks, the time to wait after the start of the container before the first one, and the number of consecutive failures needed before the kubelet takes action, so that a single transient failure does not cause a restart.

In the Event Filter case the relevant probes are liveness ones, since an instance can end up in a broken state while still running and can only recover by being restarted. More specifically, during Run 3 it has been observed that Athena applications can get stuck in unpredictable way, so having an automated restart procedure is an important upgrade. A health check of this kind was developed as part of this thesis for the Event Filter instances.

The health of each instance is reported through a *health file*⁵, whose path is passed to the container with an environment variable (figure 4.14). The file contains the current status of the instance (starting, healthy or unhealthy), while its modification time is refreshed by a periodic heartbeat. The liveness probe can then check both: a file that has not been updated for too long means that the whole process is stuck, while an unhealthy status triggers a restart immediately, without waiting for the file to become stale. To prevent the probe from reading a half-written file, the status is written atomically by first storing it to a temporary file and then moving it into place (file move is intrinsically atomic); the heartbeat, on the other hand, only updates the modification time, and never overwrites the content.

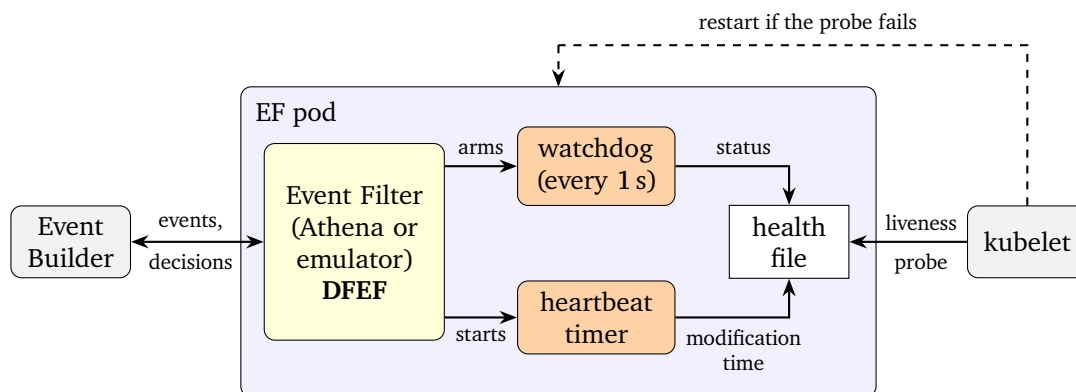


Figure 4.14: Health reporting of an Event Filter instance. The watchdog of the DFEF event provider writes the status to the health file, while the heartbeat periodically updates its modification time; the liveness probe executed by the kubelet checks both, and restarts the container when the check fails.

The health reporting is implemented in a dedicated class used by the DFEF library on the Event Filter side, allowing it to be tested interchangeably with Athena or

⁵Even if the official documentation warns that `exec` probes (what is used for checking the files) are not recommended for high pod density clusters since each `exec` call would trigger a process fork, in this case it has been evaluated feasible together with the C&C group because of the long checking times (orders of tens of seconds) and because the socket-based methods would interfere with the host network configuration.

the Event Filter emulator. Once the connection with the Event Builder is open, the instance is reported as healthy, the heartbeat is started and a watchdog checks two conditions every second:

- the connection with the Event Builder must be still open since the messaging layer never reconnects once a session is lost;
- an event handed to the Event Filter must not be held longer than a configurable timeout without a decision.

It's important to notice that the delivery time of an event is recorded when it is received from the Event Builder and not when the Event Filter asks for a new one: a pending request could simply mean that no events are being sent, for example because the trigger is on hold, so triggering an application restart would be problematic. If either of the conditions above is violated, the instance is reported as unhealthy, a final state that is never reverted.

The heartbeat period and the processing timeout are configured through environment variables as well. In this way, the probe and all the parameters it relies on are defined together in the YAML file of the deployment, and could also be propagated by the KEFController from the OKS configuration if the feature is adopted.

Chapter 5

Trigger-Level Analysis

The trigger system described in chapter 3 discards almost all the events accepted by the first level, keeping only those passing the HLT selection. Some of the rejected events, however, would still be valuable for physics: this is the case of low-mass resonances decaying into jets, produced at rates too high to fit into the bandwidth requirements of standard triggers. *Trigger-Level Analysis* (TLA) recovers part of these events by recording only the objects already reconstructed by the HLT, at a small fraction of the size of a full event and therefore at much higher rates. In this chapter, the TLA technique and its use in ATLAS are first introduced. The possible TLA workflows for the HL-LHC upgrade are then discussed, together with the tests performed to assess their impact on the dataflow system. Finally, the search for a light Z' boson decaying into quarks and produced together with a jet from initial-state radiation is studied at truth level, as an example of a physics case that would benefit from the higher rates of TLA.

5.1 Trigger-Level Analysis in ATLAS

Analyzing directly the objects reconstructed by the trigger instead of the full detector data is an idea shared by all the main LHC experiments. CMS refers to it as *data scouting*, and used it already in Run 1 to search for dijet resonances [74], while in LHCb the output of the online reconstruction is used directly for physics through the *Turbo stream* [75]. In ATLAS, the first trigger-level analysis was a search for dijet resonances on the data collected at the beginning of Run 2 [76]; the technique has since been extended to more signatures and analyses.

5.1.1 Motivation

The rate at which events can be recorded is limited by the bandwidth of the TDAQ system and by the computing resources needed to reconstruct them offline. In Run 3, the trigger menu was optimized to fit within these constraints, with a target average HLT physics output rate of 3 kHz [47]. To stay within this budget, the trigger thresholds have to be raised, losing all the events below them.

The effect is especially severe for signatures with low-momentum objects and a large Standard Model background, the most evident example being jets. As shown in figure 1.5, jet production dominates the cross-section of all the other processes and its rate grows steeply as the transverse momentum decreases. In Run 2, the lowest-threshold unprescaled single-jet trigger therefore required $p_T > 420$ GeV at the HLT level [77]. For a dijet resonance search, this threshold translates directly into a lower bound on the accessible masses. For two jets with the same transverse momentum, back-to-back in the transverse plane, the invariant mass is:

$$m_{jj} = 2p_T \cosh y^*, \quad y^* = \frac{y_1 - y_2}{2} \quad (5.1)$$

where y_1 and y_2 are the rapidities of the two jets. Since the QCD background is dominated by t -channel scattering, the two jets are often produced at large rapidity separations, while the resonance decay products remain more central. This directly translates to a common requirement for dijet searches of $|y^*| < 0.6$ which, together with the efficiency turn-on of the trigger, sets the lower edge of the fully-efficient region probable for a mass search at $m_{jj} > 1.1$ TeV during Run 2 [77]. The search for new resonances could therefore only access masses above about 1.5 TeV [78] since, to correctly identify a resonance, the full peak (or most of it) should fall into the region of maximum trigger efficiency.

The size of the loss is shown in figure 5.1, comparing the dijet mass distributions recorded in Run 2 by the standard data stream (420 GeV trigger and prescaled lower-threshold triggers) and by the TLA stream (100 GeV L1 trigger) with the same selection: as expected, below about 1 TeV, the distribution recorded by the standard stream falls behind the TLA one, which collects up to almost two orders of magnitude more events.

Similar limitations affect the searches for light resonances decaying into photons or muons and for long-lived particles, all relying on low-momentum objects [79].

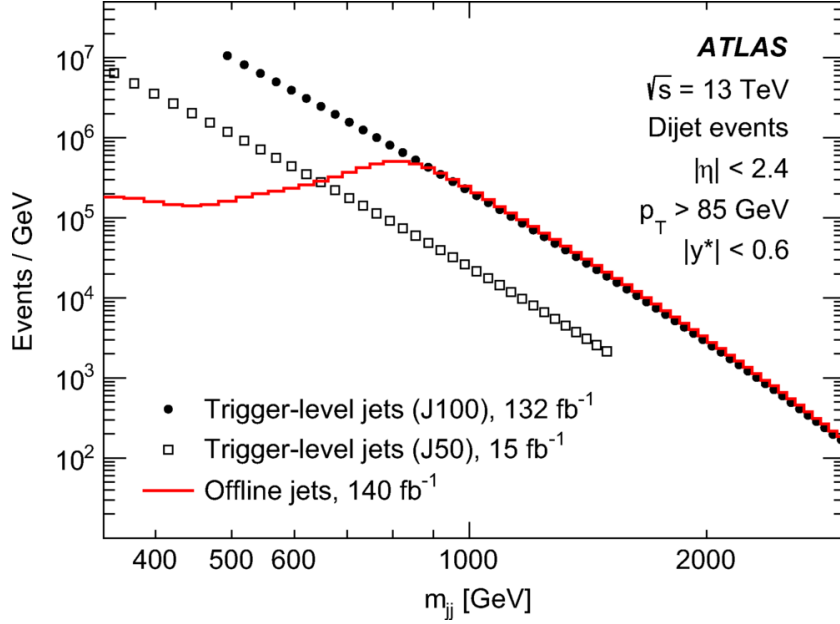


Figure 5.1: Dijet invariant mass distributions for events collected by the TLA stream and reconstructed in the HLT, with the $p_T > 100$ GeV L1 trigger (filled circles) and with the prescaled single-jet and dijet $p_T > 50$ GeV triggers (open squares), compared with the events collected by the main stream and reconstructed offline (histogram) [78].

5.1.2 The TLA Workflow

Most of the events rejected at the HLT have already been reconstructed by it, with algorithms close to the offline ones (section 3.3.3). TLA exploits this reconstruction: for the events accepted by dedicated TLA chains, which run the same algorithms as the standard ones but with much lower thresholds, only the physics objects reconstructed by the HLT are saved to the TLA stream. As anticipated in section 3.3.3, a TLA event weighed on average 4.5 kB in 2022, about 0.3 % of a full event, so that TLA chains could record events at rates of the order of 1 kHz to 10 kHz while using only a minor fraction of the HLT output bandwidth [47].

The HLT-reconstructed objects, though, are not reduced to their four-momenta. For example, in the case of jets the TLA stream also contains the variables needed to derive a dedicated calibration at the analysis level, such as the energy fractions deposited in the different calorimeters, the number of primary vertices reconstructed by the HLT and the jet area [47]. Still, the HLT objects cannot be reconstructed again offline, a strong limitation for more complex signatures. For this reason, during Run 3 the TLA event content was extended with the tracks reconstructed by the HLT, causing the addition of custom *Partial Event Building* (PEB) chains

to the TLA menu. These save the raw detector data in regions of interest around the HLT objects, which can then be fully reconstructed offline, at the price of a larger size (typically about 10 % of a full event). In 2025, the TLA chains recorded events at an average rate of 9 kHz during luminosity levelling, with an average size of 40 kB, for a total throughput of about 370 MB s^{-1} [79].

5.1.3 Results

The first ATLAS trigger-level analysis searched for dijet resonances with masses between 450 and 1800 GeV, using up to 29.3 fb^{-1} of data at $\sqrt{s} = 13 \text{ TeV}$ [76]. The search was then extended to 132 fb^{-1} of Run-2 data, with up to 75 times more events than available to conventional searches in the same mass range, where thanks to an end-of-fill triggering scheme and to L1Topo, the lower edge of the search was moved from 600 down to 375 GeV. No significant excess was observed, and limits were set on the production of new resonances decaying into two jets.

This lower edge comes from a structural limitation of the trigger. In an inclusive dijet search, the jets firing the trigger are the decay products of the resonance itself. Through (5.1), the minimum jet p_T required by the trigger corresponds to a minimum dijet mass, below which the trigger is no longer fully efficient. In this region, the m_{jj} distribution no longer follows the smoothly falling QCD spectrum, but is distorted (*sculpted*) by the efficiency turn-on of the trigger, a distortion that could easily hide or mimic a resonance. For this reason, in the sample collected with the lowest-threshold L1 trigger all the events with m_{jj} below 344 GeV had to be discarded [78].

A way around this limitation is to require the resonance to recoil against an additional jet or photon from initial-state radiation (ISR). The trigger can then fire on the ISR object instead of the decay products, which are recorded at much lower momenta since the minimum accessible mass is no longer tied to the trigger threshold. On the other hand, an ISR process yields a lower signal rate: the additional emission adds a vertex to the production diagram, corresponding to a further power of the coupling constant to its cross-section, α_s for a jet and α_{EM} for a photon [80]. Since $\alpha_{EM} \ll \alpha_s$, the suppression is much stronger for photon ISR than for jet ISR.

Depending on its momentum, the resonance can then be reconstructed in two ways. When its momentum is moderate, its decay products form two separate jets (*resolved* topology) and its mass is measured from m_{jj} . When its momentum is much larger than its mass, instead, its decay products appear collimated in a single large-radius jet (*boosted* topology, section 2.6.1), whose mass then provides

the mass of the resonance.

Both topologies were exploited in Run 2 with standard triggers: the resolved search covered masses between 200 and 650 GeV [81], while the boosted one, using photon ISR, reached masses down to 20 GeV [82]. Combining ISR with TLA allows lower trigger thresholds and higher rates, extending the sensitivity to even lower masses and couplings. This is the approach followed, in the boosted topology, by the physics case of section 5.3.

5.2 TLA Workflows in Run 4

In Run-4, the Event Filter will reconstruct all the events accepted by the L0 at 1 MHz, with tracking performed either in regions of interest or on the full detector, while only about 1 % of them will be recorded (section 3.4.3). As in Run 3, the reconstruction already performed on the rejected events could be preserved through TLA, further increasing the rates with respect to today values. In February 2025, ATLAS appointed a dedicated task force to evaluate the possible TLA scenarios for Run 4, in terms of physics potential, technical feasibility and impact on the TDAQ resources [79]. Its conclusions are the starting point of the dataflow tests carried out in this thesis.

5.2.1 Run-4 Scenarios

The most ambitious option considered by the task force was to implement TLA directly in the L0 Global Trigger, saving the calorimeter clusters and the muons reconstructed by Global at the collision rate. This *globalTLA* would bypass both the L0 and the EF selections, producing a dataset completely unbiased by the trigger. However, the task force concluded that it should not be prioritized for Run 4. The lack of tracking at L0 strongly limits the usability of the data in the high pile-up conditions of the HL-LHC, and the expected throughput of about 20 GB s^{-1} would require significant additional hardware on top of a dedicated data path between Global and the Event Aggregator, not foreseen in the dataflow architecture.

TLA in the Event Filter is instead a purely software-based workflow, with a much lower impact on the TDAQ system; for this reason, it was recommended as the priority for Run 4. Three scenarios were defined, with increasing ambition and resource usage:

- **Run-3-like:** the TLA streams of Run 3 are replicated in the EF. Scaling the 7.5 kHz of the Run-3 TLA stream with the luminosity gives a rate of the order

of 75 kHz, while the expected throughput is at most $1 \pm 0.5 \text{ GB s}^{-1}$, once only the tracks associated with physics objects or with the primary vertex are kept;

- **high-rate:** the products of the EF reconstruction are saved for all the events rejected by the EF, about 150 kHz with full-scan tracking and 250 kHz with regional tracking only, for a total of 400 kHz and a throughput of up to 5 GB s^{-1} ;
- **high-rate with PEB:** raw detector data in regions of interest are added to the previous scenario, with about 1 GB s^{-1} of additional throughput for every 2 kHz of PEB rate.

5.2.2 Impact on Dataflow

As mentioned, TLA events are partial events, similar to those of the calibration streams. In the baseline Run-4 dataflow, partial events are foreseen at about 100 kHz, with a throughput of 4.85 GB s^{-1} , alongside the 10 kHz of full events corresponding to 48.5 GB s^{-1} . Assuming the Event Aggregator as a scaled-out version of the Run-3 SFO (since the definitive architecture is yet to be decided) and since the performance of hard drives has not changed significantly in recent years, each Event Aggregator server is expected to write about 800 MB s^{-1} of events received from the Event Filter, as in Run 3, while the same amount is read back for the transfer to offline. The full farm would then require around 69 servers; bandwidth-wise, the high-rate scenario remains of the same order as the partial-event budget since most of it is saturated by full events anyway, requiring at most a few additional servers.

However, the high rate typical of TLA streams could raise a different problem. The dataflow system is designed for large events at comparatively low rates, while TLA events are small and very frequent: in the high-rate scenario, the Event Aggregator would receive 400 kHz of TLA events, four times the rate foreseen for all the partial events in the baseline dataflow requirements. At such rates, the fixed cost paid for each event becomes more important than its size. Every event has to be received over TCP, a protocol known to be CPU-intensive for high rates of small messages, and then processed and written to disk. The maximum rate of partial events sustainable by the Event Filter nodes and by the Event Aggregator had never been measured, and the task force indicated it as one of the main open points for the second phase of its work [79]. Since the Event Aggregator will be built from the same kind of hardware as the Run-3 SFO, this question can already be thoroughly investigated by performing preliminary tests on the Run-3 system, as done in this

thesis context and explained in the next section.

5.2.3 Test Setup

The tests focused on the first two scenarios of the task force, the Run-3-like and the high-rate one. The high-rate scenario was also repeated assuming a high non-TLA calibration requirement, with the calibration streams increased to 5 GB s^{-1} , to account for the case in which TLA and calibration both need a substantial share of the partial-event bandwidth.

They were performed at Point 1 at the beginning of February 2026 during the *Year-End Technical Stop* (YETS), using a Run-3 dataflow partition in which only the DCMs and the SFOs were real applications. The HLTSV generated the events internally at 100 kHz in all the tests (section 4.2.1) and the different TLA rates were obtained by changing the fraction of events accepted by the DCMs for the TLA stream. More specifically, each DCM generated event fragments of configurable size instead of requesting them from the ROS and took the selection decision with no processing time. For the tests, the DCM was modified to add a TLA stream, with configurable rate and event size, to the existing physics and calibration streams. Each event was accepted for each stream with a configurable probability; all relevant parameters were adjusted during the run to achieve the needed test conditions, including the calibration streams rate and physics acceptance. The accepted events were then sent to the SFOs and written to disk as during data taking. Finally, a dedicated script read and deleted each file once closed to emulate the transfer to permanent storage, thus subjecting the disks to the same load as in normal operation.

The SFO system used in the tests, however, is much smaller than the Event Aggregator foreseen for Run 4: with 10 servers it can write about 8 GB s^{-1} , compared with the 55 GB s^{-1} of physics and partial events that require about 69 servers in Run 4 (section 5.2.2). For this reason, all the Run-4 rates and bandwidths were divided by the ratio of the two, about 6.9. Even the 400 kHz of the high-rate scenario then correspond to less than 60 kHz, well within the 100 kHz generated by the HLTSV. The number of DCM nodes followed the same scaling: 145 nodes correspond to about 1000 Event Filter nodes sending data to the Event Aggregator. With the increased calibration, the total throughput rises to 60 GB s^{-1} , and the scaling factor to 7.5. Table 5.1 summarizes the resulting targets, together with the values measured in the tests.

Each scenario was tested in three configurations:

Table 5.1: TLA rate and bandwidth for each tested scenario: scaling factor, Run-4 value, target for the 10 SFO servers used in the tests, and average value measured at Point 1 without compression.

	Run-3-like	High-rate	High-rate, high calibration
Scaling factor	6.9	6.9	7.5
Run-4 rate [kHz]	75	400	400
Run-4 bandwidth [GB/s]	1.5	5	5
Scaled target rate [kHz]	10.9	58.2	53.3
Scaled target bandwidth [MB/s]	218	727	667
Measured rate [kHz]	11.0	58.8	52.1
Measured bandwidth [MB/s]	239	808	673

- **no compression**, with 3 DCM instances per node: events are written to disk as they are received, so this configuration shows the load caused by the event rate and size alone;
- **ZLIB compression**, with 3 DCM instances per node: the DCMs compress the events before sending them to the SFOs, moving part of the load from the storage to the processing nodes. To keep the same written bandwidth, the size of the generated fragments was increased, so that the compressed events kept the target size;
- **ZLIB compression**, with 15 DCM instances per node: the same as above, with the compression work spread over more parallel processes.

The CPU usage was monitored on one SFO server and on one DCM node, since all the machines showed a similar load.

These tests are a first, preliminary assessment of the feasibility of TLA workflows in Run-4: they rely on the Run-3 hardware and dataflow software, with emulated events, and are meant to identify the possible bottlenecks well before the Event Aggregator is built.

Table 5.2: Results of the TLA tests at Point 1 for each tested scenario and configuration: average CPU usage of an SFO server and of the DCMs on a node, where 100 % corresponds to one core, and occurrence of busy signals from the DCMs.

	Run-3-like	High-rate	High-rate, high calibration
<i>No compression, 3 DCM instances per node</i>			
SFO CPU [%]	293	338	345
DCM CPU [%]	235	272	270
Busy signals	none	none	none
<i>ZLIB compression, 15 DCM instances per node</i>			
SFO CPU [%]	291	347	358
DCM CPU [%]	426	475	470
Busy signals	none	few	few

5.2.4 Results

The results are summarized in tables 5.1 and 5.2. Without compression, the TLA rate and bandwidth reached the scaled targets in all the scenarios, with no busy signal from the DCMs. Remarkably, the CPU usage of the SFO depends only weakly on the rate of TLA events: increasing it by a factor of five, from the Run-3-like to the high-rate scenario, raised the average usage from about 2.9 to 3.4 cores, with no thread exceeding 80 %. The fixed cost per event is therefore well within the capabilities of the SFO, and the 400 kHz of the high-rate scenario could be sustained by an Event Aggregator of the planned size (figure 5.2).

Compression, on the other hand, moves the load to the DCMs. With only 3 instances per node, their threads saturated and the DCMs frequently signalled to be busy, as their output could not keep up with the incoming events. Increasing the parallelism to 15 instances per node removed most of the busy signals, and the targets were reached also in this configuration, at the price of an increase of the DCM CPU usage from about 270 to 475 %, while the SFO usage remained unchanged (figure 5.3).

Finally, the high-rate scenario was run overnight for about six and a half hours, to check the stability of the system over a long period (figure 5.4). In this run, the calibration streams were kept at their nominal rate, about 350 MB s^{-1} in total, alongside a TLA stream of about 58 kHz and 700 MB s^{-1} . The written rate and bandwidth remained stable for the whole run, and the CPU usage of the SFOs

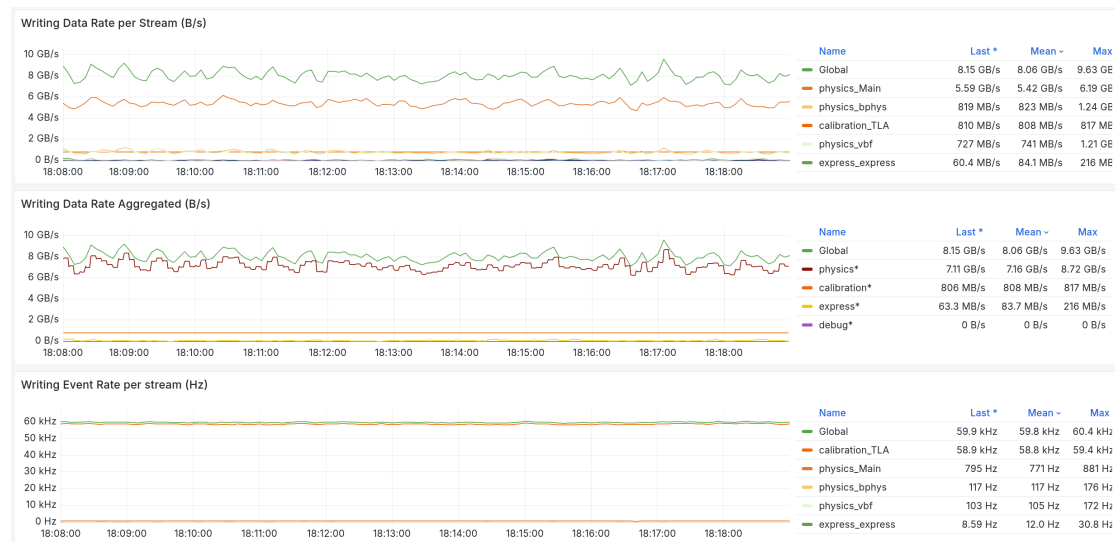


Figure 5.2: Bandwidth and event rate written by the SFOs per stream in the high-rate scenario without compression. The TLA stream reaches 58.8 kHz and about 800 MB s^{-1} , alongside about 7 GB s^{-1} of physics streams.

and of the DCMs stayed consistent with the values measured in the shorter tests. Compression was not enabled: in the shorter tests, some busy signals still appeared with 15 instances per node even when the DCM threads were not saturated, a behavior that is yet to be understood.

5.3 The Z' Benchmark Model

New resonances decaying into a quark-antiquark pair are predicted by many extensions of the Standard Model and are among the main targets of TLA. The benchmark adopted by the ATLAS dijet searches (including trigger-level ones [78]) is the Z' boson of the simplified dark matter models, a new spin-1 particle coupled to quarks. The same benchmark is used in the study presented in this chapter, so that its results can be directly compared with the current ATLAS limits.

5.3.1 Simplified Dark Matter Models

Dark matter (DM), which constitutes about five times as much of the mass of the universe as ordinary matter, has not been observed in particle physics experiments so far, and there is no evidence of any non-gravitational interaction between DM and Standard Model particles [83]. However, if DM interacts with quarks through a new mediator particle, both the DM and the mediator could be produced in

Chapter 5. Trigger-Level Analysis

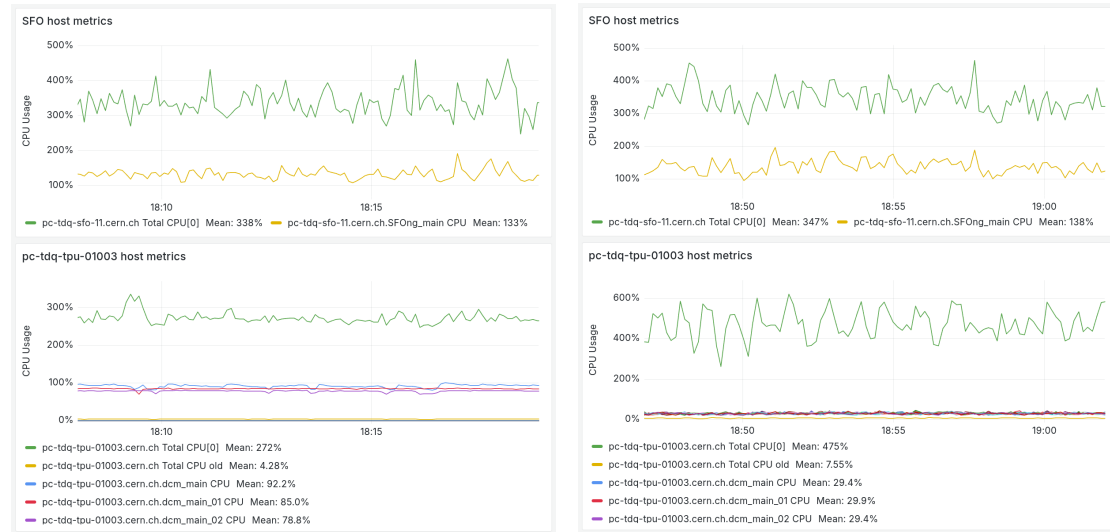


Figure 5.3: CPU usage in the high-rate scenario, without compression (left) and with ZLIB compression and 15 DCM instances per node (right), where 100 % corresponds to one core. Top: total CPU usage of an SFO server (green) and of the SFO process (yellow). Bottom: total CPU usage of a DCM node (green), the same quantity normalized to the 64 hardware threads of the node (yellow), and usage of the individual DCM instances. Without compression, each of the 3 DCM instances uses almost a full core, while with compression the load is spread over 15 instances, each using about 30 %.

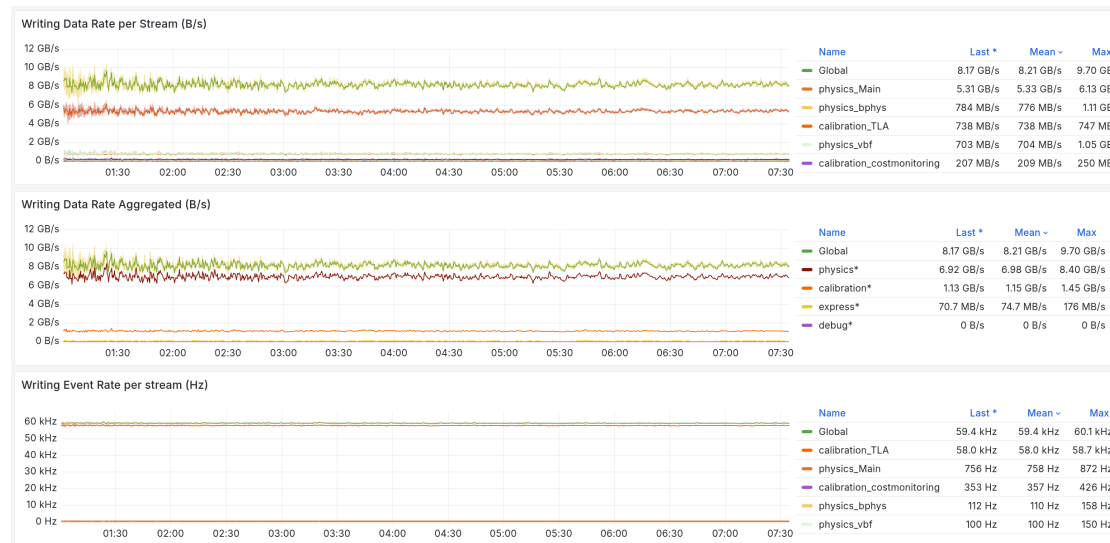


Figure 5.4: Event rate and bandwidth of the TLA stream written by the SFOs during the overnight run in the high-rate scenario, with nominal calibration and without compression.

proton-proton collisions at the LHC and the mediator could be observed through its decays back into quarks. To interpret the collider searches without relying on a complete theory, ATLAS and CMS adopted a common set of *simplified models*, which add to the Standard Model only a DM particle χ , assumed to be a Dirac fermion, and a *mediator* connecting it to quarks. In the s -channel models, each case is fully described by four parameters: the DM mass m_χ , the mediator mass $M_{Z'}$, the coupling of the mediator to quarks g_q and its coupling to DM g_χ . Furthermore, lepton couplings are set to zero to avoid the stringent limits from dilepton searches, meaning that the mediator is assumed to be *leptophobic* [84].

The ATLAS benchmark is a spin-1 mediator Z' with axial-vector couplings, described by the Lagrangian:

$$\mathcal{L}_{\text{axial}} = -g_\chi Z'_\mu \bar{\chi} \gamma^\mu \gamma^5 \chi - g_q \sum_q Z'_\mu \bar{q} \gamma^\mu \gamma^5 q \quad (5.2)$$

where the sum runs over all the quark flavors, using the same coupling g_q for all of them. This universality guarantees that the model does not introduce new sources of flavor violation, strongly constrained by flavor physics [84]. A vector mediator, obtained replacing $\gamma^\mu \gamma^5$ with γ^μ , is also defined.

Through the coupling to quarks, the Z' can be produced at the LHC by quark-antiquark annihilation, and decay back into a quark-antiquark pair, giving a dijet resonance (figure 5.5).

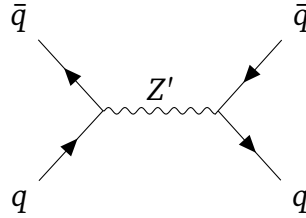


Figure 5.5: Resonant production of a Z' boson through quark-antiquark annihilation and its decay into a quark-antiquark pair. Each vertex is proportional to the coupling g_q .

More generally, the Z' can decay into any pair of quarks or DM particles that is kinematically accessible, with partial widths:

$$\Gamma_{q\bar{q}} = \frac{g_q^2 M_{Z'}}{4\pi} (1 - 4z_q)^{3/2}, \quad \Gamma_{\chi\bar{\chi}} = \frac{g_\chi^2 M_{Z'}}{12\pi} (1 - 4z_\chi)^{3/2} \quad (5.3)$$

where $z_{q,\chi} = m_{q,\chi}^2/M_{Z'}^2$, each width vanishing below the corresponding threshold $M_{Z'} < 2m_{q,\chi}$ [84]. The factor three of difference between the two denominators

comes from the quark color. For all quarks lighter than the top, $z_q \simeq 0$ and the width reduces to $g_q^2 M_{Z'}/4\pi$ per flavor, the same value obtained with vector couplings, meaning that the two models can be considered equivalent for the dijet signature.

Instead, the vector and axial-vector models differ in how DM would interact with ordinary matter, which is probed by *direct detection* experiments, looking for the recoil of nuclei scattered by the DM particles of the galactic halo. Depending on the mediator, this scattering can be:

- **Spin-independent**, for vector couplings: the DM particle couples to all the nucleons of the nucleus, whose contributions add up coherently, so that the cross-section grows with the square of the number of nucleons. Since direct detection experiments often use heavy nuclei such as xenon, they are extremely sensitive to this case and their limits largely exceed those from colliders.
- **Spin-dependent**, for axial-vector couplings: the DM particle couples to the spin of the nucleus, to which nucleons in pairs with opposite spins do not contribute, losing the coherent enhancement. Therefore, the direct detection limits are several orders of magnitude weaker, making collider searches more competitive [84]. This is exactly the reason why the axial-vector mediator is the benchmark used by ATLAS to present the results of the dijet searches.

The comparison between the two cases is shown in figure 5.6.

In the benchmark used by ATLAS to present the dijet results, the DM coupling is set to $g_\chi = 1$, while the DM mass is set to the very large value $m_\chi = 10$ TeV [85]. The decay into a DM pair would require $M_{Z'} > 2m_\chi = 20$ TeV, far above all the mediator masses accessible at the LHC: according to (5.3), the width $\Gamma_{\chi\bar{\chi}}$ would vanish making the Z' decay only into quarks. For $g_q = 0.1$ and five light flavors, its total width would then be about 0.4 % of the mass: the Z' is therefore a narrow resonance, with a width negligible compared with the experimental resolution.

In this scenario, both the production rate and the width depend only on g_q and $M_{Z'}$ (at leading order) and the production rate scales quadratically with g_q [78]. A limit on the cross-section can therefore be translated directly into a limit on g_q , the quantity reported by the ATLAS dijet searches; current ATLAS limits are summarized in figure 5.7.

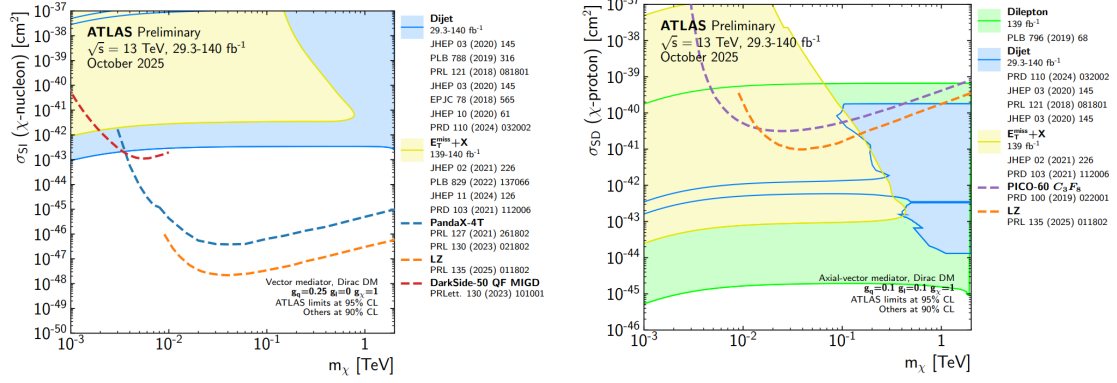


Figure 5.6: ATLAS exclusions on the DM-nucleon scattering cross-section (shaded regions) compared with the limits from direct detection experiments (dashed lines), in the spin-independent case for a vector mediator (left) and in the spin-dependent case for an axial-vector mediator (right). The direct detection limits reach $\approx 10^{-48} \text{ cm}^2$ in the first case and $\approx 10^{-41} \text{ cm}^2$ in the second, where the collider searches exclude a large part of the remaining parameter space. Since the conversion of the collider results depends on the couplings, each panel refers to a specific benchmark scenario [85].

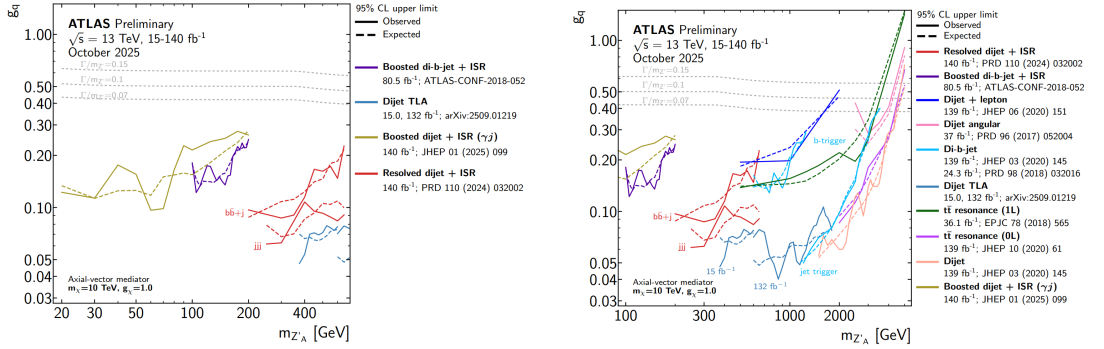


Figure 5.7: Upper limits at 95 % confidence level on the coupling g_q of an axial-vector Z' mediator as a function of its mass, set by the ATLAS searches for dijet resonances with Run-2 data, for $m_\chi = 10 \text{ TeV}$ and $g_\chi = 1$, in the mass ranges 20–700 GeV (left) and 90–5500 GeV (right). The dotted lines show the values of g_q corresponding to fixed ratios of width to mass [85].

5.3.2 Production with Initial-State Radiation

As shown in figure 5.7, below the 375 GeV reached by the dijet TLA analysis all the ATLAS limits come from searches requiring an additional ISR jet or photon, and below 100 GeV only from the boosted search with photon ISR [82]. Requiring a jet instead of a photon gives a much larger signal rate (section 5.1.3), but the

multijet background increases the trigger rate as well. Standard jet triggers can only sustain this rate with high thresholds, which can be lowered significantly using TLA workflows (as mentioned in section 5.1.1).

The study presented in this chapter therefore focuses on jet ISR. To describe how the Z' would be produced at the LHC, it is necessary to consider the hard scattering process to be happening between proton's constituents, the *partons*, each carrying a fraction x of the proton momentum. Following the factorization theorem, the cross-section of a process can be written as [86]:

$$\sigma(pp \rightarrow X) = \sum_{i,j} \int dx_1 dx_2 f_i(x_1, \mu_F^2) f_j(x_2, \mu_F^2) \hat{\sigma}_{ij \rightarrow X}(\hat{s}, \mu_R^2, \mu_F^2) \quad (5.4)$$

where $f_i(x, \mu_F^2)$ are the *parton distribution functions* (PDFs) of the proton, $\hat{\sigma}_{ij \rightarrow X}$ is the partonic cross-section, computed perturbatively in powers of α_s , and $\hat{s} = x_1 x_2 s$ is the squared center-of-mass energy of the partonic collision. The arbitrary scales μ_R and μ_F are the renormalization and factorization scales. Roughly speaking, the latter separates the two parts of the calculation: emissions with transverse momenta above μ_F are included in the partonic cross-section, while those below are absorbed into the PDFs.

At leading order, the direct production of figure 5.5 would require $\hat{s} = M_{Z'}^2$, so only parton pairs with $x_1 x_2 = M_{Z'}^2/s$ would contribute. For a light Z' this product is tiny, about 1×10^{-5} for 50 GeV at $\sqrt{s} = 14$ TeV. However, the resulting decay quarks would carry a transverse momentum of at most $M_{Z'}/2$, far below any jet trigger threshold.

At the next order in α_s , new partonic channels open, so that the Z' could be produced together with an additional parton, with $\hat{s} \geq M_{Z'}^2$. The corresponding processes are $q\bar{q} \rightarrow Z'g$ and $qg \rightarrow Z'q$ (or $\bar{q}g \rightarrow Z'\bar{q}$), shown in figure 5.8. In the first case the ISR jet comes from a gluon, in the second from a quark. Either way, once the Z' decayed, the final state would contain three partons: the quark-antiquark pair and the ISR one.

Besides the additional power of α_s discussed in section 5.1.3, the rate of this process strongly depends on the momentum of the ISR parton. This follows from the general QCD property stating that when an emitted gluon becomes soft or collinear to the parton emitting it, the matrix element factorizes as [86]:

$$d\Phi_n | \mathcal{M}_n |^2 \simeq d\Phi_{n-1} | \mathcal{M}_{n-1} |^2 \frac{\alpha_s C_i}{\pi} \frac{d\theta^2}{\theta^2} \frac{dE}{E} \quad (5.5)$$

where $\mathcal{M}_n$ and $d\Phi_n$ are the matrix element and the phase space with n partons, E is the energy of the gluon, θ its angle with respect to the emitting parton and

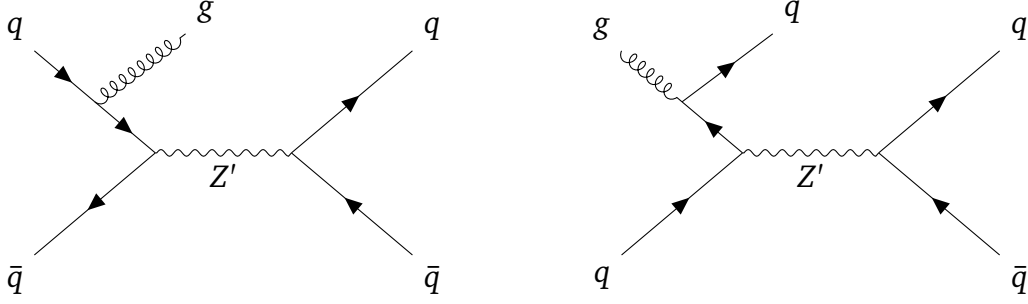


Figure 5.8: Examples of leading-order diagrams for the production of a Z' in association with an ISR parton, through $q\bar{q} \rightarrow Z'g$ (left) and $qg \rightarrow Z'q$ (right), followed by the decay of the Z' into a quark-antiquark pair.

C_i the color factor of the latter, $C_F = 4/3$ for quarks and $C_A = 3$ for gluons. The emission probability diverges for $E \rightarrow 0$ and $\theta \rightarrow 0$, meaning that most of the radiation is emitted with low energy or at small angles (i.e. close to the beam axis in the ISR case); this directly implies that only a small fraction of the events would contain an ISR parton with a large transverse momentum.

Moreover, a hard ISR would significantly raise the energy needed for the partonic collision. For a Z' recoiling against a massless parton, both with transverse momentum p_T , the squared partonic center-of-mass energy is:

$$\hat{s} = M_{Z'}^2 + 2p_T^2 + 2p_T m_T \cosh \Delta y, \quad m_T = \sqrt{M_{Z'}^2 + p_T^2} \quad (5.6)$$

where Δy is the rapidity difference between the two. Its minimum, reached for $\Delta y = 0$, is $\sqrt{\hat{s}_{min}} = m_T + p_T$. For $p_T = 120$ GeV, this amounts to about 242, 250 and 276 GeV for a Z' of 20, 50 and 100 GeV respectively. When the mass is small compared with the transverse momentum, $m_T \simeq p_T$ and $\sqrt{\hat{s}_{min}} \simeq 2p_T$: the minimum energy of the partonic collision is then set by the transverse momentum of the recoil, almost independently of the mass of the Z' . The CMS search for boosted resonances between 50 and 300 GeV with Run-2 data is an example of this: due to the bandwidth limitations of standard triggers, the resonance was required to have $p_T > 500$ GeV [87], corresponding to $\sqrt{\hat{s}_{min}} \approx 1$ TeV for a Z' of 50 GeV, four times the value obtained with $p_T = 120$ GeV.

5.3.3 Boosted Decay and Jet Substructure

In the boosted topology, the decay products of the Z' would be reconstructed as a single large-R jet (section 2.6.1), with the same transverse momentum as the ISR jet at leading order, meaning that a threshold on the leading jet would apply to

both. The single-jet TLA trigger emulated for Run 4, used in this study, requires a leading jet with $p_T > 120$ GeV, the value used as an example in section 5.3.2.

Whether the two quarks actually end up in the same jet depends on how the momentum of the Z' is shared between them. For two partons carrying fractions z and $1 - z$ of the transverse momentum and separated by a small angle ΔR , the invariant mass is [33]:

$$m^2 \simeq z(1 - z) p_T^2 \Delta R^2 \quad \Rightarrow \quad \Delta R \simeq \frac{m}{p_T \sqrt{z(1 - z)}} \quad (5.7)$$

The separation is minimal for symmetric decays ($z = 1/2$), giving the approximation $\Delta R \approx 2m/p_T$ of section 2.6.1, and grows for asymmetric ones. For a Z' of 50 GeV with $p_T = 120$ GeV, ΔR goes from about 0.8 for $z = 1/2$ to 1.4 for $z = 0.1$, where the softer quark is no longer contained in a jet of radius $R = 1.0$. A Z' of 100 GeV, instead, would need $p_T \gtrsim 200$ GeV to be contained even with a symmetric decay. The fraction of boosted events therefore decreases quickly as the mass increases.

A large- R jet initiated by a single quark or gluon also has a mass, generated by the emissions inside the jet. According to (5.5), the emission probability grows as the energy and the angle of the emitted gluon decrease, so that most emissions are soft or collinear; as shown by (5.7), these add very little mass to the jet. The mass of QCD jets is therefore typically small compared with their transverse momentum, and its distribution falls smoothly towards large values after a peak at low masses (the *Sudakov peak*) [88]. Moreover, due to their large area, these jets also collect soft radiation uncorrelated with the hard process, coming from ISR, pile-up and the *underlying event* (the interactions between the remnants of the colliding protons). This radiation adds mass to both signal and background jets, degrading the resolution.

Among the grooming algorithms mentioned in section 2.6.1, the one used in this study is *soft drop* [33]. The constituents of the jet are first reclustered with the Cambridge/Aachen algorithm ($p = 0$ in (2.4)), which merges them in order of angular distance only, producing an angular-ordered clustering tree. The tree is then traversed backwards, starting from the widest-angle step:

- the last clustering step is undone, splitting the jet into two subjects j_1 and j_2 ;
- if the two subjects satisfy the *soft drop condition*

$$\frac{\min(p_{T,1}, p_{T,2})}{p_{T,1} + p_{T,2}} > z_{cut} \left(\frac{\Delta R_{12}}{R} \right)^\beta \quad (5.8)$$

the procedure stops, and the current jet is the groomed one;

- otherwise, the subjet with lower p_T is removed and the procedure is repeated on the other one.

Here z_{cut} is the soft threshold and β the angular exponent. For $\beta = 0$ only the momentum fraction is tested, while for $\beta \rightarrow \infty$ the jet is left ungroomed. The jets used in this study are groomed with $z_{cut} = 0.1$ and $\beta = 1$.

For a boosted Z' , the soft wide-angle radiation would be removed in the first steps, until the declustering reaches the branching separating the two quarks. With $\beta = 1$, the condition requires a momentum fraction above $z_{cut} \Delta R/R$, which is always below $z_{cut} = 0.1$ for two quarks contained in the jet. Decays asymmetric enough to fail it would mostly not be contained in the first place: for the 50 GeV example above, containment alone requires $z \gtrsim 0.22$. Both quarks are then kept and the mass of the Z' is preserved. On the other hand, in QCD jets the widest-angle branchings are mostly soft emissions failing the condition, whose removal reduces the jet mass. As a result, soft drop pushes the mass of quark and gluon jets towards lower values, while preserving the mass of jets coming from resonances decaying into a quark-antiquark pair (figure 5.9) [87].

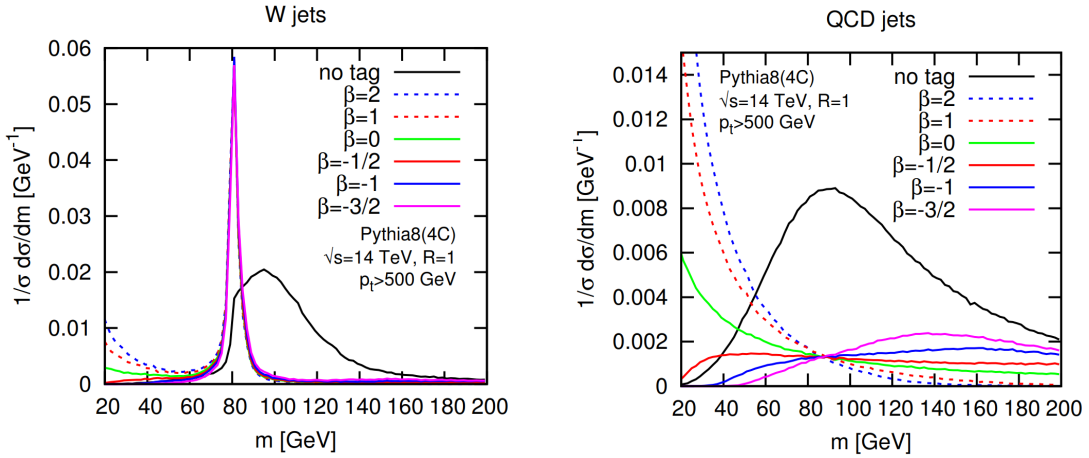


Figure 5.9: Mass distributions of jets originating from boosted W bosons decaying into a quark-antiquark pair (left) and of QCD jets (right), with $R = 1.0$ and $p_T > 500$ GeV, simulated with Pythia 8 in proton-proton collisions at 14 TeV, before (no tag) and after soft drop with different values of β . For each β , z_{cut} is chosen to keep 35 % of the W jets in the mass window between 70 and 90 GeV. The signal remains a narrow peak around the W mass, while the QCD distribution, which before grooming accidentally peaks in the same region, is pushed away from the signal [33].

Beyond the mass, the internal structure of the jet can also be exploited directly. The radiation of a Z' jet would be concentrated around two hard cores, one for each quark (*two-prong* structure), while QCD jets usually have a single hard core surrounded by soft radiation. Substructure observables such as the N-subjettiness ratio τ_{21} or the energy correlation ratio D_2 measure how compatible the jet is with two hard prongs rather than one [88]. Alternatively, specific machine learning taggers can be used to identify the two-prong structure, as the CMS search mentioned in section 5.3.2.

5.3.4 Backgrounds

Since the signal consists only of jets, any Standard Model process producing a large- R jet together with other jets can mimic it. By far the dominant background is QCD multijet production, whose cross-section exceeds that of all the other processes at the LHC (figure 1.5). In these events, the leading large- R jet usually originates from a single quark or gluon, and its mass is generated by the emissions described in section 5.3.3. After grooming, its distribution is smooth and steeply falling, and a signal would appear as a localized excess on top of it. However, a single hard emission at a sizable angle is enough to give a QCD jet a mass comparable to that of the Z' , together with a two-prong structure. Even if such configurations are rare, multijet events are produced at such a high rate that they remain the largest contribution in any mass window.

The production of W and Z bosons in association with jets, followed by their decay into a quark-antiquark pair, is instead a *resonant* background. It proceeds through the same diagrams of figure 5.8, with the Z' replaced by a W or a Z boson, and gives exactly the signature of the signal, with peaks at about 80 and 91 GeV. In the CMS search mentioned in section 5.3.2, these peaks are clearly visible above the multijet background (figure 5.10). Top-quark pair production contributes a smaller, non-resonant background, mainly in final states with b -quarks, while Higgs boson, single-top and diboson production are minor.

In the study presented in this chapter, only the multijet background is simulated.

5.4 Truth-Level Sensitivity Study

The physics case of section 5.3 was studied with simulated events, as a preliminary study of the possible usage of TLA data for the boosted Z' signature in Run 4. More specifically, the study aims at estimating the sensitivity that the lower thresholds of TLA could make accessible, and at understanding which aspects of the signature

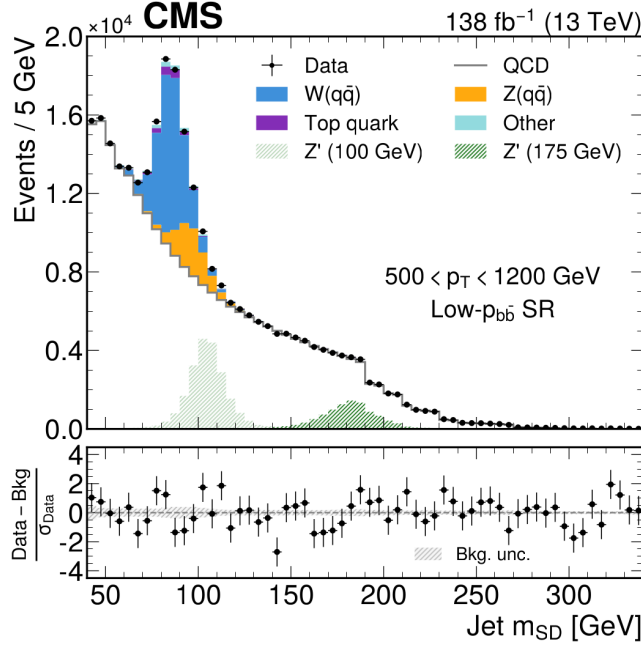


Figure 5.10: Example of soft-drop mass distribution of the large-R jet in the CMS search for boosted resonances produced with ISR, in the signal region targeting decays into light or charm quarks, using 138 fb^{-1} of Run-2 data, with the background-only fit. The W and Z peaks are visible on top of the multijet background, while the expected signals of a Z' of 100 and 175 GeV with $g_q = 0.25$ are shown for comparison. The steplike features come from a requirement on $\rho = 2 \ln(m_{SD}/p_T)$, resulting in a different mass range in each p_T bin [87].

would require further work in view of a dedicated analysis. For this purpose, the signal is studied at truth level, while the multijet background is fully simulated and reconstructed.

5.4.1 Samples and Selection

The signal samples were generated with the benchmark of section 5.3.1: an axial-vector Z' with $g_q = 0.1$, $g_\chi = 1$ and $m_\chi = 10 \text{ TeV}$, produced together with an ISR jet and decaying into a quark-antiquark pair. The matrix element was computed with MadGraph5_aMC@NLO [89], while the parton shower and the hadronization were simulated with Pythia 8 [90]. Eleven mass points were produced, from 20 to 350 GeV, with 5000 events each. At generator level, at least one jet was required to have $p_T > 50 \text{ GeV}$ and a further filter kept only the events with a jet above 85 GeV; the fraction of events passing the filter ε_{filter} was computed by MadGraph together

with the cross-section of the process. No detector simulation was performed for the signal, since the analysis uses only truth jets (section 2.6.3), built with the anti- k_t algorithm with $R = 1.0$ and groomed with soft drop, with $z_{cut} = 0.1$ and $\beta = 1$.

The multijet background was simulated with Pythia 8 at $\sqrt{s} = 14$ TeV, including the full detector simulation and an average pile-up of $\langle\mu\rangle = 140$, corresponding to the nominal HL-LHC conditions (section 1.3.3). Since the cross-section falls by many orders of magnitude with the jet momentum, the events are generated in five slices of the leading truth-jet p_T (JZ0–JZ4, from 0 to 800 GeV), with 200 000 events each. Each slice is then weighted to the expected number of events:

$$w = \frac{\sigma \varepsilon_{filter} \mathcal{L}_{int}}{\sum_i w_i} \quad (5.9)$$

where σ and ε_{filter} are the cross-section and the filter efficiency of the slice, $\mathcal{L}_{int}$ the integrated luminosity and $\sum_i w_i$ the sum of the generator weights of the processed events. Furthermore, if the leading jet comes from a pile-up interaction rather than from the hard scattering, the event is removed since its large weight would appear as an isolated spike in the spectrum.

Unlike the signal ones, background large-R jets are reconstructed from *Unified Flow Objects* (UFOs), which combine the particle flow inputs of section 2.6.1 with track-cluster objects optimized for high transverse momenta [91]. More specifically, the pile-up contribution is reduced before clustering with *Constituent Subtraction*, subtracting from each input its expected pile-up energy based on the event energy density, followed by *SoftKiller*, which removes the inputs below a p_T threshold chosen event by event. The jets are then groomed with soft drop as for the signal, but are not calibrated. Each reconstructed jet is also required to match a truth jet within $\Delta R < 0.75$. The reasons for this choice of jet definition are discussed in section 5.4.4.

Both samples are then selected with the same requirements. The trigger is emulated as the single-jet TLA trigger considered by the ATLAS TLA group as a candidate for Run 4, requiring the leading large-R jet to have $p_T > 120$ GeV and $|\eta| < 2.5$. Each event must then contain at least two large-R jets with $p_T > 20$ GeV and $|\eta| < 2.5$, one for the Z' and one for the ISR. The Z' candidate is then searched for in both the leading and the subleading large-R jet, defining two exclusive categories that are combined at the end of the analysis (section 5.4.5). In both cases, the observable of the search is the soft-drop mass m_J of the candidate jet.

5.4.2 Signal Kinematics

Before using jets, the signal was studied at parton level¹ to understand the process and give preliminary insights on expected analysis results. More specifically, a Z' is considered *merged* when the angular separation of its decay quarks is $\Delta R(q, \bar{q}) < 1.0$, equivalent to the radius of the large-R jets, meaning that both quarks should be contained in a single jet.

As discussed in section 5.3.2, the transverse momentum of the Z' is set by its recoil against the ISR jet. In these samples, it is further shaped by the generator filter and by the trigger, but not by the mass: after the emulated Run-4 trigger, the median p_T of the Z' is about 150 GeV for all the masses between 20 and 100 GeV. Through (5.7), the typical separation of the decay quarks therefore grows linearly with the mass, from about 0.3 at 20 GeV to 0.7 at 50 GeV and 1.3 at 100 GeV, which is where the distributions of figure 5.11 peak. Among the events passing the trigger, the fraction of merged Z' drops accordingly from 95 % at 20 GeV to 65 % at 50 GeV and only 12 % at 100 GeV.

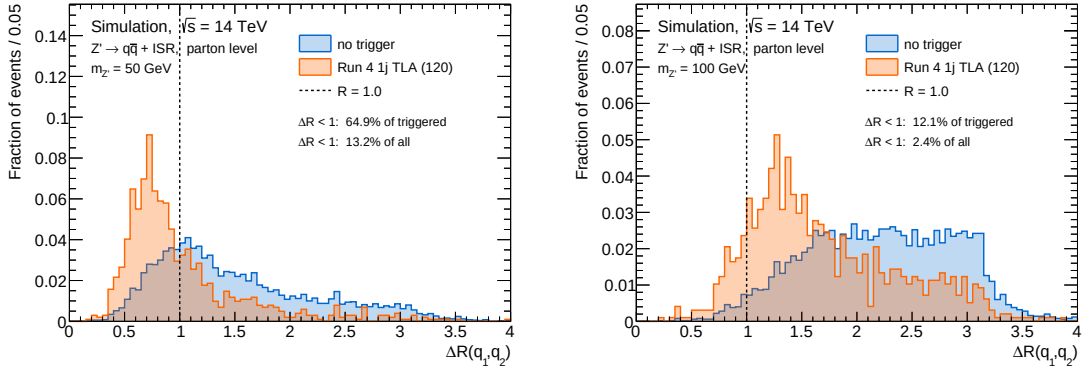


Figure 5.11: Angular separation of the two quarks from the Z' decay for $m_{Z'} = 50$ GeV (left) and 100 GeV (right), for all the generated events and for those passing the emulated Run-4 single-jet TLA trigger. The dashed line marks the radius of the large-R jets.

A simple upper bound on the merged fraction follows from the same relation. A symmetric decay is merged only if $p_T > 2m$, while asymmetric decays require an even larger momentum (5.7): the fraction of events with $p_T(Z') > 2m_{Z'}$ is therefore the largest merged fraction that the samples can provide. As shown in figure 5.12, the measured fraction indeed lies below this bound at all masses, by up to a factor of two (27 % against 50 % at 50 GeV). More importantly, the bound itself falls steeply with the mass, below 5 % at 100 GeV and 1 % at 150 GeV,

¹Here *studied at partone level* means identifying the two quarks from the decay of the Z' in the truth record and analyze their properties.

meaning that above about 100 GeV these samples cannot provide a usable boosted signal due to the small statistics. This is a property of the samples rather than of the boosted topology, which at higher masses requires a Z' momentum growing with the mass, while in these samples it stays fixed.

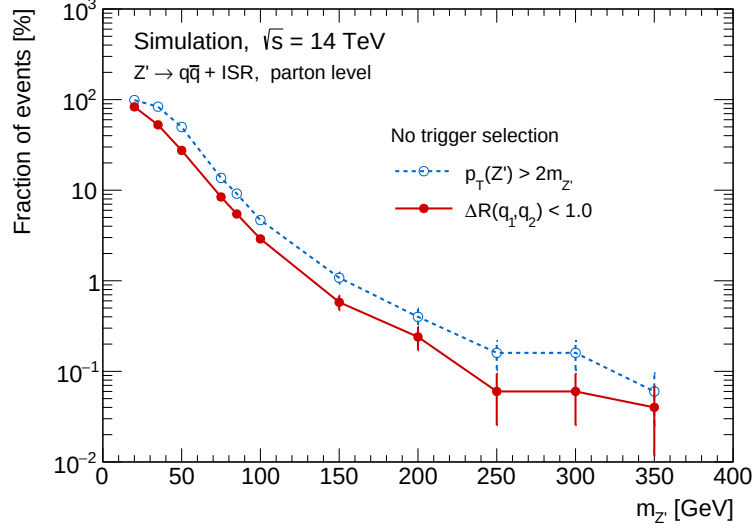


Figure 5.12: Fraction of generated signal events with $p_T(Z') > 2m_{Z'}$, an upper bound on the merged fraction, compared with the measured fraction of events with $\Delta R(q, \bar{q}) < 1.0$, as a function of the Z' mass, before any trigger requirement.

The trigger threshold affects the merged signal in two opposite ways, shown in figure 5.13 for the triggers of Run 2 and Run 3 and for the two candidates for Run 4. A higher threshold selects Z' with larger momenta, so the merged fraction among the selected events increases; for example, with the Run-2 single-jet trigger, emulated with a threshold of 475 GeV, almost all the selected Z' are merged up to 100 GeV. However, the same threshold rejects almost all the events, so that the merged events passing the trigger are less than 0.1 % of those generated. With the Run-4 single-jet TLA trigger, instead, about 15 % of all the generated events between 20 and 50 GeV are both selected and merged, around four times more than with the TLA triggers of Run 2 and Run 3.

5.4.3 Signal Reconstruction

At jet level, a merged Z' should appear as a single large-R jet with a mass close to $m_{Z'}$. To understand how the signal is reconstructed, two categories of jets are defined using the truth information: a jet is *matched* to the Z' if its axis lies within $\Delta R < 0.75$ of the Z' direction, and *contained* if, in addition, both decay quarks lie

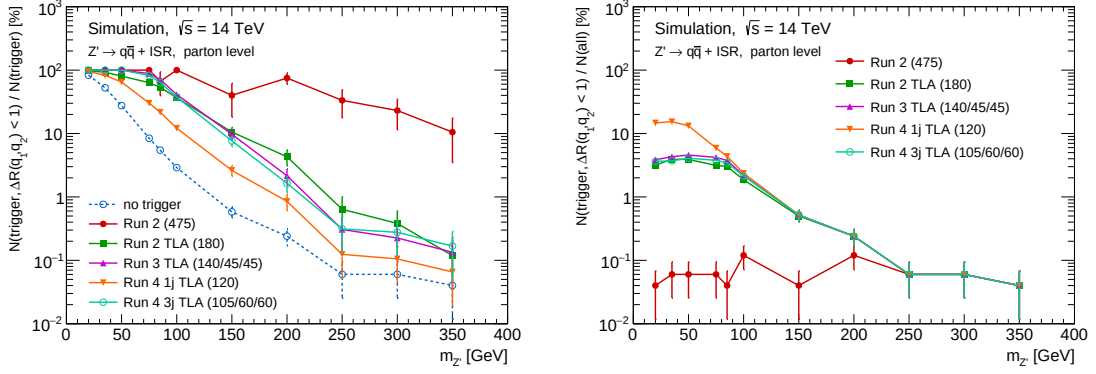


Figure 5.13: Fraction of merged Z' as a function of the mass for different trigger emulations, relative to the events passing each trigger (left) and to all the generated events (right). For the trigger requiring three objects, the thresholds are applied to the three partons of the event.

within $\Delta R < 1.0$ of its axis. These categories are only used to study the signal, and do not enter the selection.

As shown in figure 5.14, only the contained jets form a peak at the mass of the Z' , with a tail towards higher values due to the additional radiation collected by the jet. The jets that point at the Z' without containing it hold only one of the two quarks; their mass is therefore the one of a single-parton jet, small compared with their transverse momentum (section 5.3.3), and they accumulate at low masses.

The same figure shows how often the contained jet is the leading large- R jet of the event. At leading order, the Z' and the ISR jet have the same transverse momentum, so that either of the two can be the leading one after the parton shower: at 20 GeV, the Z' is the leading jet in only 56 % of the events. The fraction grows to about 74 % at 75 GeV, consistently with the containment requirement selecting Z' with $p_T > 2m$ (section 5.4.2), which are more often harder than the ISR. In the remaining events, the Z' is reconstructed as the subleading jet and would be lost by a search using the leading jet only. For this reason, a second category has been defined for this study, in which the Z' candidate is the subleading jet. The two categories are kept exclusive, so that an event can enter the subleading category only if the mass of its leading jet is not compatible with the signal (section 5.4.5).

5.4.4 Background

The multijet background is simulated in five slices of the leading truth-jet transverse momentum: 0–20, 20–60, 60–160, 160–400 and 400–800 GeV for JZ0 to JZ4

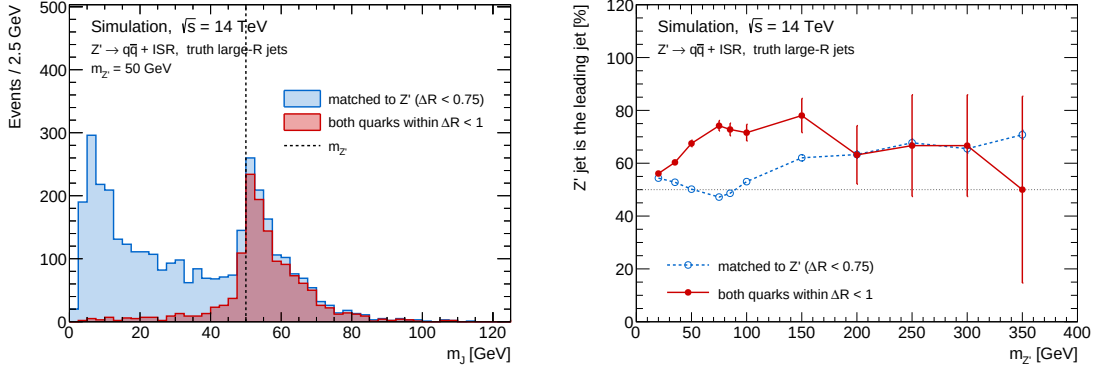


Figure 5.14: Left: mass of the large-R jet matched to the Z' (blue) and of the jets also containing both decay quarks (red), for $m_{Z'} = 50$ GeV. Right: fraction of events in which the matched (blue) or contained (red) jet is the leading large-R jet, as a function of the Z' mass.

respectively. After the selection of section 5.4.1, the lowest slices are almost entirely rejected by the trigger, in fact no JZ0 event passes and only a single JZ1 event does, contributing 0.1 % of the total yield. The selected fraction then rises to 14 % for JZ2, 81 % for JZ3 and 98 % for JZ4.

The choice of the large-R jet definition for the background required particular care. A first version of the study used large-R jets built from topo-clusters at the electromagnetic scale (*EMTopo*), groomed with soft drop but without any pile-up subtraction or calibration. The resulting background spectrum showed an unexpected peak at about 75 GeV and the UA2 fit described below failed completely. To understand the origin of the problem, the leading large-R jet of the same events was reconstructed with *EMTopo* inputs, with *UFO* inputs and at truth level, and the three were compared after the same selection (table 5.3 and figure 5.15).

The comparison shows two distinct problems of the *EMTopo* jets:

- at low transverse momentum, the large jet area collects a large amount of pile-up energy, which is not subtracted, causing soft JZ1 and JZ2 jets to be pushed above the trigger threshold with an inflated momentum and mass and producing the spurious peak (up to 73 GeV for JZ1);
- at high transverse momentum, the missing calibration starts dominating, since the calorimeters are non-compensating (section 2.6.1), the *EMTopo* jets measure only about 80 % of the truth momentum.

The *UFO* jets with CS+SK pile-up mitigation solve both problems: their momentum

Table 5.3: Comparison of the leading large-R jet reconstructed with EMTopo and UFO inputs and at truth level, on the same background events with $\langle\mu\rangle = 200$: number of events passing the selection, average ratio of the reconstructed to the truth p_T , and average difference between the reconstructed and the truth mass.

	Passing events	p_T^{reco}/p_T^{truth}	$m^{reco} - m^{truth}$ [GeV]
	EMTopo / UFO / truth	EMTopo / UFO	EMTopo / UFO
JZ1	31 / 0 / 0	2.34 / –	+73 / –
JZ2	832 / 760 / 1190	1.04 / 1.08	+43 / +1.7
JZ3	2637 / 2405 / 2598	0.80 / 1.01	+12 / –1.8
JZ4	2963 / 2938 / 2952	0.77 / 1.00	–11 / –1.2

is within 8 % of the truth one and their mass within about 2 GeV, with no JZ1 event passing the selection. For these reasons the background was moved to UFO jets, producing the final samples with $\langle\mu\rangle = 140$, corresponding to the nominal HL-LHC conditions.

The final background spectrum is shown in figure 5.16. Contrary to the dijet mass of the resolved topology, the mass of the leading large-R jet does not fall smoothly; in fact, after a peak at low masses, it shows a shoulder between about 40 and 65 GeV, dominated by JZ2, and then changes slope around 150 GeV (where JZ3 and JZ4 take over). This shape is a consequence of how QCD jets acquire their mass (section 5.3.3): since the mass comes from the radiation inside the jet, its distribution is peaked, with the peak moving to higher values as the jet momentum grows. The trigger then cuts the jet p_T sharply at 120 GeV, so the many JZ2 jets with p_T between 120 and 160 GeV all place their mass peak in the same region, creating the shoulder. In the resolved topology, the corresponding trigger footprint would lie at much lower dijet masses, below the region of the search, while here it falls inside it.

The background spectrum of a resonance search is usually modelled with a smooth function fitted to the data. Following the dijet searches, a three-parameter UA2 function² was used:

$$f(x) \propto (1 - x)^{p_1} x^{p_2 + p_3 \ln x}, \quad x = \frac{m_J}{\sqrt{s}} \quad (5.10)$$

fitted between 15 and 150 GeV. As shown in figure 5.16, the function cannot follow the shoulder, with $\chi^2/\text{ndf} \approx 10$, and underestimates the background in

²The name comes from the UA2 experiment at the CERN SPS proton-antiproton collider, which used a similar falling function of x , with four parameters, to describe the dijet mass spectrum [92].

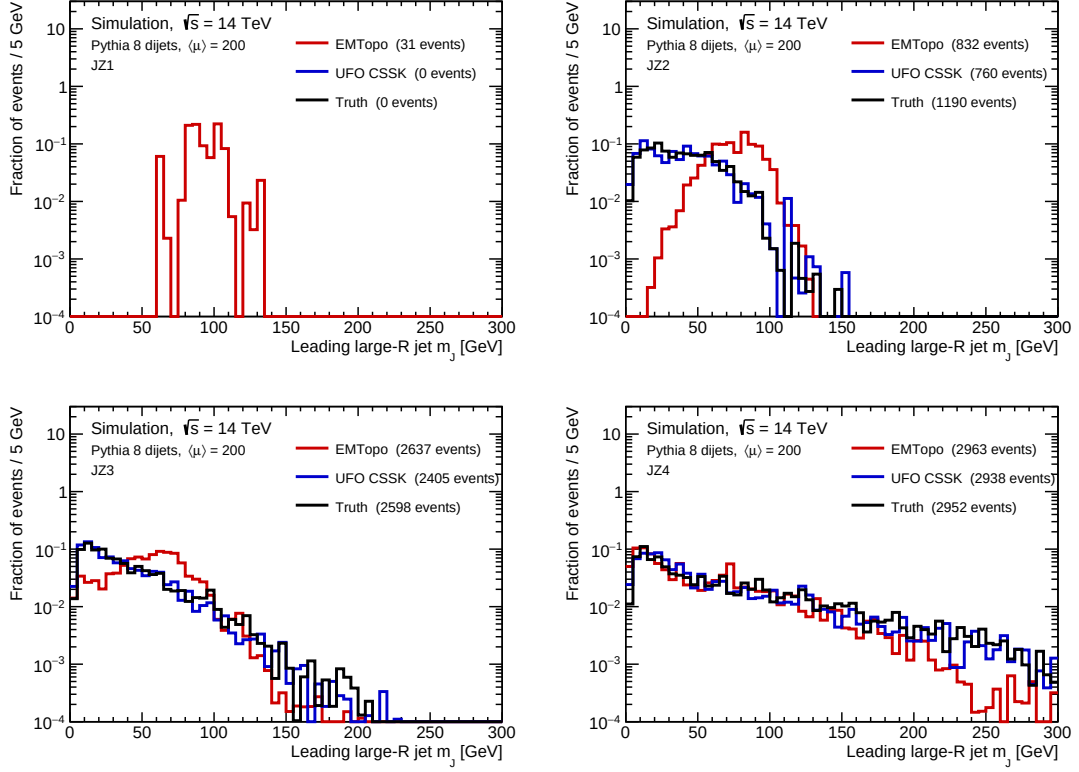


Figure 5.15: Mass of the leading large-R jet after the selection for the JZ1 to JZ4 slices, reconstructed with EMTopo (red) and UFO (blue) inputs and at truth level (black), on the same background events with $\langle\mu\rangle = 200$. Each distribution is normalized to unit area.

the signal regions: compared with the direct count of the simulated events, the fit is 29 % lower at 20 GeV and between 11 and 16 % lower at 50, 75 and 85 GeV. For this reason, the background in each signal region is taken directly from the weighted count of the simulated events rather than from the fit. With 200 000 events per slice, the statistical uncertainty on this count ranges from 1.5 to 4.6 % across the signal regions of both categories.

5.4.5 Sensitivity Estimate

For each mass point, the signal region is defined as a window around the peak of the Z' candidate mass. The distribution of m_J for the signal is fitted with a *double-sided Crystal Ball* (DSCB) function, made of a Gaussian core and two power-law

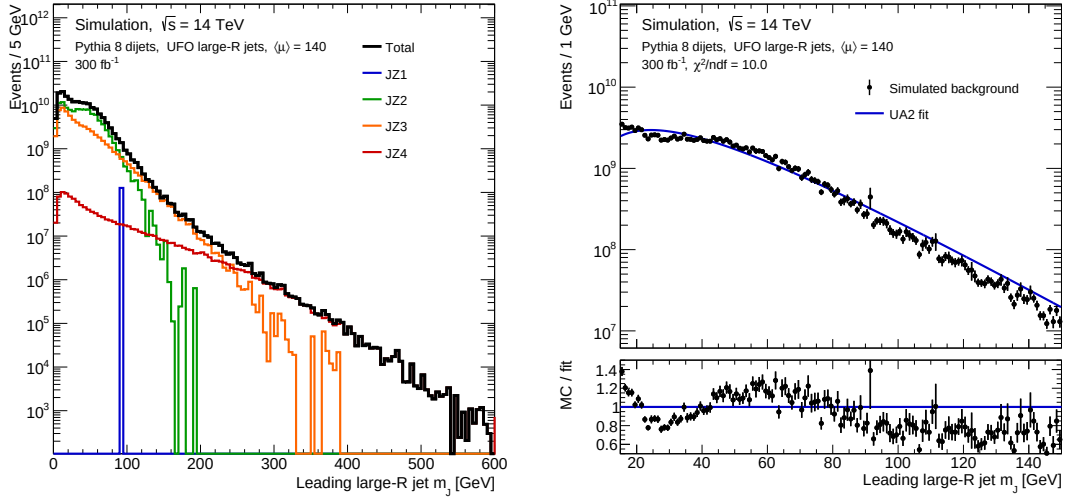


Figure 5.16: Left: mass of the leading large-R jet of the multijet background after the selection, for each JZ slice and in total, normalized to 300 fb^{-1} . Right: fit of the total spectrum with the UA2 function between 15 and 150 GeV.

tails:

$$f(t) \propto \begin{cases} \left(\frac{n_L}{\alpha_L}\right)^{n_L} e^{-\alpha_L^2/2} \left(\frac{n_L}{\alpha_L} - \alpha_L - t\right)^{-n_L} & t < -\alpha_L \\ e^{-t^2/2} & -\alpha_L \leq t \leq \alpha_H \\ \left(\frac{n_H}{\alpha_H}\right)^{n_H} e^{-\alpha_H^2/2} \left(\frac{n_H}{\alpha_H} - \alpha_H + t\right)^{-n_H} & t > \alpha_H \end{cases} \quad t = \frac{m_J - \mu}{\sigma} \quad (5.11)$$

where μ and σ are the mean and the width of the core, while $\alpha_{L,H}$ and $n_{L,H}$ define where the low and high tails start and how fast they decrease. The tails account for the asymmetric shape of the peak, with the high-mass tail coming from the additional radiation collected by the jet (section 5.4.3). The window is then defined as $\mu \pm 2\sigma$, as shown in figure 5.17 for $m_{Z'} = 50 \text{ GeV}$. The fit is well constrained up to 50 GeV, with a relative uncertainty on σ below 16 %, while at 75 and 85 GeV the lower number of merged events raises it to about 35 %. Above 100 GeV, no usable peak is formed (section 5.4.2), and these masses are not considered further.

The expected number of signal events in the window is:

$$S = \sigma \varepsilon_{\text{filter}} \mathcal{L}_{\text{int}} (A \times \varepsilon) \quad (5.12)$$

where σ and $\varepsilon_{\text{filter}}$ are the cross-section and the filter efficiency computed by MadGraph, $\mathcal{L}_{\text{int}}$ the integrated luminosity and $A \times \varepsilon$ the fraction of the generated

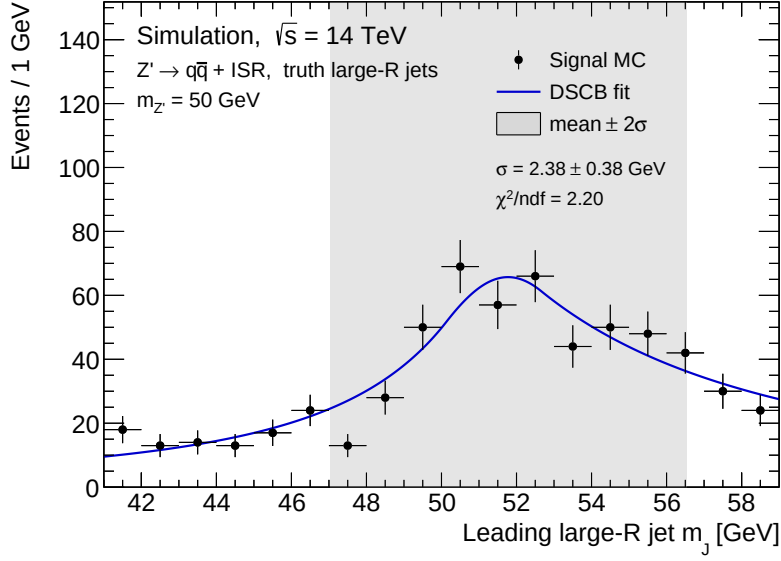


Figure 5.17: DSCB fit of the mass of the leading large-R jet for the signal with $m_{Z'} = 50$ GeV. The resulting signal window is 47.0 GeV to 56.5 GeV.

events that pass the selection with m_j inside the window. For the leading jet, $A \times \varepsilon$ ranges between 4 and 9 % up to 75 GeV, and drops to 1.9 % at 85 GeV, where fewer Z' are merged. All the yields are computed for $\mathcal{L}_{\text{int}} = 300 \text{ fb}^{-1}$, a small fraction of the 3000 fb^{-1} expected at the end of the HL-LHC program (section 1.3.1). The background B in the same window is instead the weighted count of the simulated multijet events (section 5.4.4).

The sensitivity is estimated with the median significance expected for a signal S over a background B , in the asymptotic approximation [93]:

$$Z = \sqrt{2 \left[(S + B) \ln \left(1 + \frac{S}{B} \right) - S \right]} \quad (5.13)$$

which reduces to $Z \simeq S/\sqrt{B}$ for $S \ll B$, the case of this study, where S/B is of the order of 10^{-3} . The expected limit at 95 % confidence level corresponds to the signal yield S_{95} giving $Z = 1.645$. Since the yield scales linearly with the cross-section, the limit on the cross-section times branching ratio follows as $\sigma \times BR = \sigma S_{95}/S$. The limit is purely statistical and no systematic uncertainty is included.

Finally, the leading and subleading categories of section 5.4.3 are combined. An event enters the leading category (L) if the mass of the leading jet falls inside its

window, and the subleading category (S) if the mass of the subleading jet falls inside its own window while the leading one does not, so that double counting is avoided; the windows of the subleading category are then derived from a separate DSCB fit of the subleading jet mass. Treating the two categories as independent, their significances are combined in quadrature, $Z^2 = Z_L^2 + Z_S^2$, and the combined limit is again the cross-section giving $Z = 1.645$.

The resulting expected limits are shown in figure 5.18 and reported in table 5.4, together with the cross-section of the benchmark model. The combination with the subleading category improves the limit of the leading category alone by 8 to 15 %, recovering part of the events in which the Z' is not the leading jet. The limits are almost constant up to 50 GeV and then become weaker with the mass, following the fraction of merged Z' (section 5.4.2).

Table 5.4: Expected limits at 95 % confidence level on $\sigma \times BR$ for $\mathcal{L}_{\text{int}} = 300 \text{ fb}^{-1}$, with the leading category only and with the combination of the leading and subleading categories, compared with the cross-section of the benchmark model with $g_q = 0.1$. The cross-section includes the generator-level requirements of section 5.4.1.

$m_{Z'}$ [GeV]	Leading [pb]	Combined [pb]	Benchmark, $g_q = 0.1$ [pb]
20	30.2	25.8	1744
35	33.3	29.0	1482
50	30.0	26.6	1293
75	45.2	41.8	1188
85	66.3	57.1	1214

These limits proved to be much stronger than the current ATLAS results. In the same mass range, the existing searches exclude couplings of about $g_q \approx 0.1\text{--}0.25$ (figure 5.7), corresponding, since $\sigma \propto g_q^2$, to cross-sections comparable to or larger than the benchmark one. Part of this difference comes from the simplifications of the study. The signal is described at truth level, without any detector resolution, and the limits only include the statistical uncertainty: with S/B of the order of 10^{-3} , even a small systematic uncertainty on the background would weaken them significantly. Moreover, the trigger was emulated by applying the 120 GeV threshold to the leading large-R jet, while the actual trigger configuration for this signature in Run 4 is yet to be defined.

On the other hand, the low threshold of the TLA trigger gives access to a region of phase space that is out of reach for the existing searches, based on photon ISR or

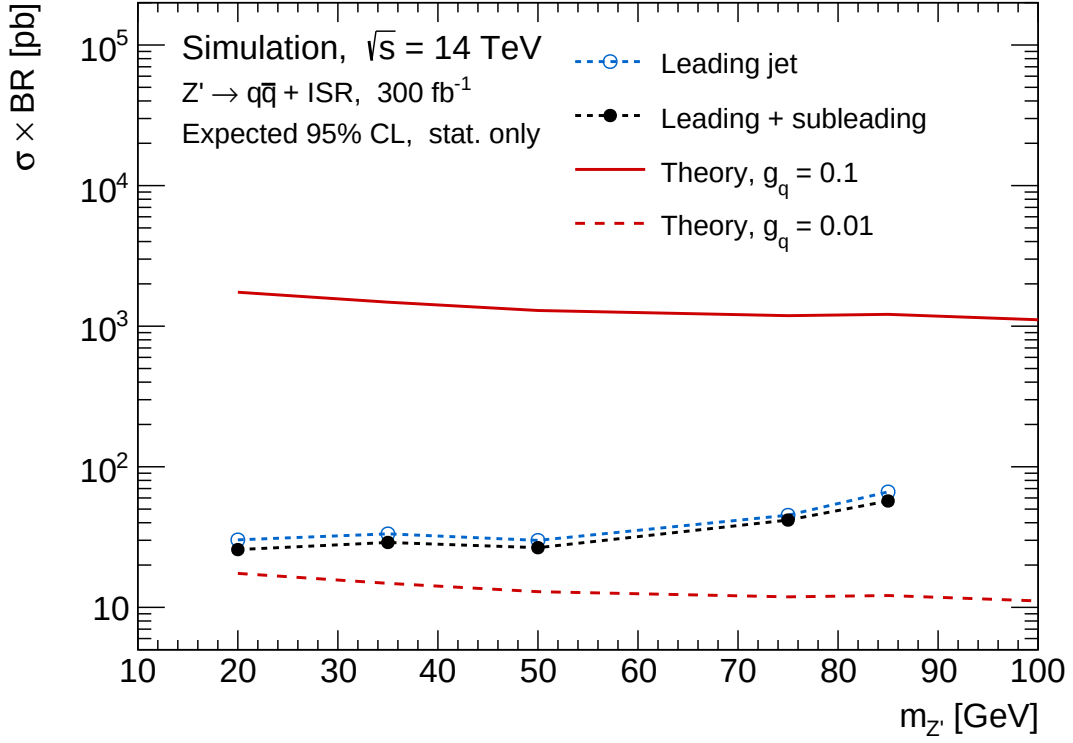


Figure 5.18: Expected limits at 95 % confidence level on $\sigma \times BR$ as a function of the Z' mass, for $\mathcal{L}_{\text{int}} = 300 \text{ fb}^{-1}$, with the leading category only (blue) and with the combination of the leading and subleading categories (black), compared with the cross-section of the benchmark model for $g_q = 0.1$ (red) and $g_q = 0.01$ (dashed red).

on much higher jet thresholds. A full physics analysis of this signature on Run-3 TLA data, already using much lower jet trigger thresholds with respect to Run 2, would probably provide a more realistic baseline to compare this preliminary analysis to, but it has yet to be started. Nevertheless, the obtained results indicate that TLA data could greatly enhance the sensitivity to low-mass boosted Z' resonances produced with jet ISR, otherwise inaccessible with the standard jet trigger thresholds, motivating further studies in view of Run 4.

Conclusion

In the first part of this thesis, a general overview of the LHC complex and the ATLAS experiment was provided, along with a complete and detailed description of its TDAQ system. The unprecedented scale of the experiment, built to operate under extreme conditions, is necessary to exploit the full physics potential of the LHC: the sole purpose of LHC experiments is to select and store the largest possible amount of physically “interesting” events occurring in the collisions, discarding the rest.

More specifically, the entire system has to be designed maximizing selection efficiency while not exceeding the available resources, such as storage, bandwidth and computing power. For this reason, a multi-level trigger system is implemented, where each selection level reduces the event rate while progressively refining the selection algorithms. The first trigger level is fully hardware-based, minimizing latencies, while the following levels leverage the flexibility of software-based algorithms running on dedicated computing farms.

The HL-LHC related upgrades to the ATLAS TDAQ system will further improve its performance, allowing it to cope with the target increase in event rate and size while also increasing the rate of events saved to permanent storage. To achieve this, the dataflow system has to be redesigned and validated at full scale well before the start of Run 4, ensuring the correct support for new workflows, such as the ones of Trigger-Level Analysis (TLA).

From March to August 2026, during my research internship at CERN, I was able to take part in the development for the HL-LHC upgrade of the ATLAS dataflow system, i.e. the TDAQ component collecting event fragments from the detector readout, delivering them to the trigger and transferring accepted events to permanent storage. I started by developing a new input source for the Run-3 version of the system, allowing already upgraded subdetectors to begin testing their new readout system for the HL-LHC (FELIX) with a complete data

acquisition chain; this new source was successfully used by the Liquid Argon Calorimeter group in its test setup. Then I moved on to the containerization (i.e. packaging applications in isolated and self-contained execution environments) and deployment of some dataflow components, testing the new setup in a full-scale joint test campaign between the dataflow and readout groups. These tests proved to be very successful, validating the new dataflow design by sustaining the Run-4 target event rate, while highlighting several issues of the infrastructure to be worked on next. Nevertheless, these results represent a first, fundamental demonstration of the containerized approach for the dataflow system, which I later complemented by developing the software needed to monitor the health of the containerized applications.

Finally, together with the ATLAS TLA group, I tested the impact of the planned upgrades to TLA workflows on the dataflow system, using the existing Run-3 infrastructure. We managed to demonstrate that even in the most demanding scenario, the targets were reached with only a modest increase in CPU usage, showing that the fixed cost of small and frequent events remains well within the capabilities of the available hardware. On the physics side, the truth-level study I carried out on the decay of a light Z' boson produced with initial-state radiation, in the boosted topology, showed promising results. More specifically, the analysis indicated that candidate Run-4 TLA trigger thresholds could collect several times more boosted signal events with respect to the Run-3 ones, with expected limits much stronger than the current ones. Although preliminary, these results strongly motivate further studies in view of Run 4, the first step being a full analysis of this physics signature on Run-3 TLA data.

Bibliography

- [1] NobelPrize.org. *The Nobel Prize in Physics 2013*. 2013. URL: http://www.nobelprize.org/nobel_prizes/physics/laureates/2013/.
- [2] Ewa Lopienska. “The CERN accelerator complex layout in 2026”. In: (2026). General Photo. URL: <https://cds.cern.ch/record/2953421>.
- [3] Michael Benedikt et al. *LHC Design Report*. CERN Yellow Reports: Monographs. Geneva: CERN, 2004. DOI: 10.5170/CERN-2004-003-V-3. URL: <https://cds.cern.ch/record/823808>.
- [4] R Bailey and Paul Collier. *Standard Filling Schemes for Various LHC Operation Modes*. Tech. rep. Geneva: CERN, 2003. URL: <https://cds.cern.ch/record/691782>.
- [5] R. Bailey. “An Application for Research: the Large Hadron Collider”. In: *CAS - CERN Accelerator School: Ion Sources*. 2013, pp. 565–574. DOI: 10.5170/CERN-2013-007.565. arXiv: 1404.0966 [physics.acc-ph].
- [6] Jean-Luc Caron. “Layout of the LEP tunnel including future LHC infrastructures.” AC Collection. Legacy of AC. Pictures from 1992 to 2002. 1997. URL: <https://cds.cern.ch/record/841560>.
- [7] Serge Dailier. “Cross section of LHC dipole.” AC Collection. Legacy of AC. Pictures from 1992 to 2002. 1999. URL: <https://cds.cern.ch/record/842530>.
- [8] Andreas Hoecker. “Physics at the LHC Run-2 and Beyond. Physics at the LHC Run-2 and Beyond”. In: (2017). Lecture notes from the 2016 European School of High-Energy Physics, 15-28 June 2016, Skeikampen, Norway (61 pages, 56 figures), pp. 153–212. DOI: 10.23730/CYRSP-2017-005.153. arXiv: 1611.07864. URL: <https://cds.cern.ch/record/2236645>.
- [9] ATLAS Collaboration. *Luminosity Public Results Run 3*. URL: <https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LuminosityPublicResultsRun3> (visited on 09/21/2026).
- [10] O. Aberle et al. *High-Luminosity Large Hadron Collider (HL-LHC): Technical design report*. CERN Yellow Reports: Monographs. Geneva: CERN, 2020.

- DOI: 10.23731/CYRM-2020-0010. URL: <https://cds.cern.ch/record/2749422>.
- [11] Andrea Dainese et al. *Report on the Physics at the HL-LHC, and Perspectives for the HE-LHC*. Tech. rep. Geneva, Switzerland, 2019. DOI: 10.23731/CYRM-2019-007. URL: <https://cds.cern.ch/record/2703572>.
 - [12] CERN. *About HiLumi LHC*. URL: <https://hilumilhc.web.cern.ch/about-hilumi-lhc/> (visited on 09/21/2026).
 - [13] The ATLAS Collaboration. “The ATLAS Experiment at the CERN Large Hadron Collider”. In: *Journal of Instrumentation* 3.08 (Aug. 2008), S08003. DOI: 10.1088/1748-0221/3/08/S08003. URL: <https://dx.doi.org/10.1088/1748-0221/3/08/S08003>.
 - [14] Riccardo Maria Bianchi and ATLAS Collaboration. “ATLAS experiment schematic or layout illustration”. General Photo. 2022. URL: <https://cds.cern.ch/record/2837191>.
 - [15] Daniel Dominguez, Kevin Patrice Moles, and Sascha Mehlhase. “ATLAS detector schematics”. In: (2021). URL: <https://cds.cern.ch/record/2777214>.
 - [16] ATLAS Collaboration. “Experiment Briefing: Keeping the ATLAS Inner Detector in perfect alignment”. General Photo. 2020. URL: <https://cds.cern.ch/record/2723878>.
 - [17] Dave Robinson. “ATLAS Tracker and Pixel Operational Experience”. In: *PoS Vertex 2016* (2017), p. 005. DOI: 10.22323/1.287.0005.
 - [18] Joao Pequenaio. “Computer generated images of the Pixel, part of the ATLAS inner detector.” 2008. URL: <https://cds.cern.ch/record/1095925>.
 - [19] *Technical Design Report for the ATLAS Inner Tracker Pixel Detector*. Tech. rep. Geneva: CERN, 2017. DOI: 10.17181/CERN.F0ZZ.ZP3Q. URL: <https://cds.cern.ch/record/2285585>.
 - [20] *Technical Design Report: A High-Granularity Timing Detector for the ATLAS Phase-II Upgrade*. Tech. rep. Geneva: CERN, 2020. URL: <https://cds.cern.ch/record/2719855>.
 - [21] Michele Livan and Richard Wigmans. “Misconceptions about Calorimetry”. In: *Instruments* 1 (Apr. 2017). DOI: 10.3390/instruments1010003.
 - [22] Joao Pequenaio. “Computer Generated image of the ATLAS calorimeter”. 2008. URL: <https://cds.cern.ch/record/1095927>.
 - [23] *ATLAS Liquid Argon Calorimeter Phase-II Upgrade*. Tech. rep. Geneva: CERN, 2017. DOI: 10.17181/CERN.6QIO.YGH0. URL: <https://cds.cern.ch/record/2285582>.
 - [24] Joao Pequenaio. “Computer generated image of the ATLAS Muons subsystem”. 2008. URL: <https://cds.cern.ch/record/1095929>.

- [25] ATLAS Collaboration. *New Small Wheel Technical Design Report*. Tech. rep. Geneva: CERN, 2013.
- [26] The ATLAS Collaboration. “The ATLAS experiment at the CERN Large Hadron Collider: a description of the detector configuration for Run 3”. In: *Journal of Instrumentation* 19.05 (May 2024), P05063. DOI: 10.1088/1748-0221/19/05/P05063. URL: <https://doi.org/10.1088/1748-0221/19/05/P05063>.
- [27] Georges Aad et al. “Luminosity determination in pp collisions at $\sqrt{s} = 13$ TeV using the ATLAS detector at the LHC”. In: *Eur. Phys. J. C* 83.10 (2023), p. 982. DOI: 10.1140/epjc/s10052-023-11747-w. arXiv: 2212.09379 [hep-ex].
- [28] ATLAS Collaboration. “Performance of the ATLAS track reconstruction algorithms in dense environments in LHC Run 2”. In: *Eur. Phys. J. C* 77 (2017), p. 673. arXiv: 1704.07983.
- [29] ATLAS Collaboration. “Reconstruction of primary vertices at the ATLAS experiment in Run 1 proton–proton collisions at the LHC”. In: *Eur. Phys. J. C* 77 (2017), p. 332. arXiv: 1611.10235.
- [30] ATLAS Collaboration. “Topological cell clustering in the ATLAS calorimeters and its performance in LHC Run 1”. In: *Eur. Phys. J. C* 77 (2017), p. 490. arXiv: 1603.02934.
- [31] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. “The anti- k_t jet clustering algorithm”. In: *JHEP* 04 (2008), p. 063. arXiv: 0802.1189.
- [32] David Krohn, Jesse Thaler, and Lian-Tao Wang. “Jet Trimming”. In: *JHEP* 02 (2010), p. 084. arXiv: 0912.1342.
- [33] Andrew J. Larkoski et al. “Soft Drop”. In: *JHEP* 05 (2014), p. 146. arXiv: 1402.2657.
- [34] ATLAS Collaboration. “Jet reconstruction and performance using particle flow with the ATLAS Detector”. In: *Eur. Phys. J. C* 77 (2017), p. 466. arXiv: 1703.10485.
- [35] ATLAS Collaboration. “Jet energy scale and resolution measured in proton–proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector”. In: *Eur. Phys. J. C* 81 (2021), p. 689. arXiv: 2007.02645.
- [36] ATLAS Collaboration. “Performance of pile-up mitigation techniques for jets in pp collisions at $\sqrt{s} = 8$ TeV using the ATLAS detector”. In: *Eur. Phys. J. C* 76 (2016), p. 581. arXiv: 1510.03823.
- [37] ATLAS Collaboration. “Electron and photon performance measurements with the ATLAS detector using the 2015–2017 LHC proton–proton collision data”. In: *JINST* 14 (2019), P12006. arXiv: 1908.00005.

- [38] ATLAS Collaboration. “Muon reconstruction and identification efficiency in ATLAS using the full Run 2 pp collision data set at $\sqrt{s} = 13$ TeV”. In: *Eur. Phys. J. C* 81 (2021), p. 578. arXiv: 2012.00578.
- [39] ATLAS Collaboration. “Performance of missing transverse momentum reconstruction with the ATLAS detector using proton–proton collisions at $\sqrt{s} = 13$ TeV”. In: *Eur. Phys. J. C* 78 (2018), p. 903. arXiv: 1802.08168.
- [40] ATLAS Collaboration. “The ATLAS Simulation Infrastructure”. In: *Eur. Phys. J. C* 70 (2010), p. 823. arXiv: 1005.4568.
- [41] S. Agostinelli et al. “GEANT4 – a simulation toolkit”. In: *Nucl. Instrum. Meth. A* 506 (2003), p. 250.
- [42] Monika Wielers. *Introduction to Trigger and Data Acquisition*. May 5, 2016. URL: https://indico.cern.ch/event/518474/contributions/1198683/attachments/1268107/1878054/TDAQ_RAL_Lecture_040516.pdf.
- [43] Francesca Pastore. *Strategies and future trends for Trigger and DAQ systems in LHC experiments*. Lecture at the International School of Trigger and Data Acquisition (ISOTDAQ 2019). 2019. URL: <https://indico.cern.ch/event/739424/> (visited on 10/01/2026).
- [44] Andrea Negri. *Introduction to and Data Acquisition*. Apr. 3, 2019. URL: <https://indico.cern.ch/event/739424/contributions/3052187/attachments/1673750/2981890/isotdaq19.Negri.DaqIntro.pdf>.
- [45] *Technical Design Report for the Phase-II Upgrade of the ATLAS TDAQ System*. Tech. rep. Geneva: CERN, 2017. DOI: 10.17181/CERN.2LBB.4IAL. URL: <https://cds.cern.ch/record/2285584>.
- [46] R. Luz. “FELIX, the ATLAS readout system: from LHC Run 3 to Run 4”. English. In: *Proc. ICALEPCS2025* (Chicago, IL, USA, Sept. 20–26, 2025). International Conference on Accelerator and Large Experimental Physics Control Systems 20. JACoW Publishing, Geneva, Switzerland, Sept. 2025, pp. 1346–1350. ISBN: 978-3-95450-255-4. DOI: 10.18429/JACoW-ICALEPCS2025-THBG002. URL: <https://indico.jacow.org/event/86/contributions/10074>.
- [47] ATLAS Collaboration. “The ATLAS trigger system for LHC Run 3 and trigger performance in 2022”. In: *Journal of Instrumentation* 19.06 (June 2024), P06029. DOI: 10.1088/1748-0221/19/06/P06029. URL: <https://doi.org/10.1088/1748-0221/19/06/P06029>.
- [48] The ATLAS Collaboration. “Performance of the ATLAS muon triggers in Run 2”. In: *Journal of Instrumentation* 15.09 (Sept. 2020), P09015. DOI: 10.1088/1748-0221/15/09/P09015. URL: <https://doi.org/10.1088/1748-0221/15/09/P09015>.
- [49] Imma Riu and on behalf of the ATLAS Collaboration. “The ATLAS Level-1 Topological Trigger performance in Run 2”. In: *Journal of Physics: Conference*

- Series 898.3* (Oct. 2017), p. 032037. DOI: 10.1088/1742-6596/898/3/032037. URL: <https://doi.org/10.1088/1742-6596/898/3/032037>.
- [50] S. Artz et al. “Upgrade of the ATLAS Central Trigger for LHC Run-2”. In: *Journal of Instrumentation* 10.02 (Feb. 2015), p. C02030. DOI: 10.1088/1748-0221/10/02/C02030. URL: <https://dx.doi.org/10.1088/1748-0221/10/02/C02030>.
 - [51] The ATLAS Collaboration. *The ATLAS Collaboration Software and Firmware*. Tech. rep. All figures including auxiliary figures are available at <https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-SOFT-PUB-2021-001>. Geneva: CERN, 2021. URL: <https://cds.cern.ch/record/2767187>.
 - [52] The ATLAS Collaboration. *DAQ approved plots*. URL: <https://twiki.cern.ch/twiki/bin/view/AtlasPublic/ApprovedPlotsDAQ>.
 - [53] Collaboration ATLAS. *Technical Design Report for the Phase-II Upgrade of the ATLAS Trigger and Data Acquisition System - Event Filter Tracking Amendment*. Tech. rep. Geneva: CERN, 2022. DOI: 10.17181/CERN.ZK85.5TDL. URL: <https://cds.cern.ch/record/2802799>.
 - [54] Viacheslav Filimonov et al. *Global Trigger Versatile Module for ATLAS Phase-II upgrade*. Tech. rep. Geneva: CERN, 2022. URL: <https://cds.cern.ch/record/2843296>.
 - [55] Matías Bonaventura, Fabrice Le Goff, and Giuseppe Avolio. *ATLAS Dataflow System: Preliminary Design Review (PDR)*. EDMS document 2898809. Version 1. CERN, 2023. URL: <https://edms.cern.ch/document/2898809/1>.
 - [56] Veronique Boisvert et al. *Report of the Event Filter Technology Choice Committee*. Tech. rep. First upload is just the cover page as a placeholder. Geneva: CERN, 2026. URL: <https://cds.cern.ch/record/2956712>.
 - [57] Andrei Kazarov et al. *The Controls and Configuration Software of the ATLAS Data Acquisition System: evolution towards LHC Run 3*. Tech. rep. Geneva: CERN, 2021. DOI: 10.1051/epjconf/202125104019. URL: <https://cds.cern.ch/record/2772943>.
 - [58] R. Jones et al. “The OKS persistent in-memory object manager”. In: *IEEE Transactions on Nuclear Science* 45.4 (1998), pp. 1958–1964. DOI: 10.1109/23.710971.
 - [59] Object Management Group. *Common Object Request Broker Architecture (CORBA) Specification, Version 3.3*. 2012. URL: <https://www.omg.org/spec/CORBA/3.3>.
 - [60] Soloviev, Igor et al. “ATLAS Operational Monitoring Data Archival and Visualization”. In: *EPJ Web Conf.* 245 (2020), p. 01020. DOI: 10.1051/epjconf/202024501020. URL: <https://doi.org/10.1051/epjconf/202024501020>.

- [61] G Anders et al. *Intelligent operations of the data acquisition system of the ATLAS experiment at the LHC*. Tech. rep. 1. Geneva: CERN, 2015. DOI: 10.1088/1742-6596/608/1/012007. URL: <https://cds.cern.ch/record/1950361>.
- [62] The ATLAS collaboration. “Operation of the ATLAS trigger system in Run 2”. In: *Journal of Instrumentation* 15.10 (Oct. 2020), P10004. DOI: 10.1088/1748-0221/15/10/P10004. URL: <https://doi.org/10.1088/1748-0221/15/10/P10004>.
- [63] Andrei Kazarov, Giuseppe Avolio, and Alina Corso-Radu. “Experience with Shifter Assistant”. In: (2018). URL: <https://cds.cern.ch/record/2632671>.
- [64] Martin Fowler. *Patterns of Enterprise Application Architecture*. Addison-Wesley, 2002. ISBN: 0321127420.
- [65] Erich Gamma et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994. ISBN: 0201633612.
- [66] A. Borga et al. “The ATLAS Readout System for LHC Runs 2 and 3”. In: *Journal of Instrumentation* 18.08 (Aug. 2023), P08022. DOI: 10.1088/1748-0221/18/08/P08022. URL: <https://doi.org/10.1088/1748-0221/18/08/P08022>.
- [67] The FELIX Team. *FELIX Phase-II Firmware Specifications*. AT2-DQ-ES-0006. Version 1.099. Sept. 1, 2026. URL: https://atlas-project-felix.web.cern.ch/atlas-project-felix/user/docs/FELIX_Phase2_firmware_specs.pdf (visited on 10/09/2026).
- [68] Corso Radu, Alina et al. “Evolution of the ATLAS TDAQ online software framework towards Phase-II upgrade: Use of Kubernetes as an orchestrator of the ATLAS Event Filter computing farm”. In: *EPJ Web Conf.* 337 (2025), p. 01047. DOI: 10.1051/epjconf/202533701047. URL: <https://doi.org/10.1051/epjconf/202533701047>.
- [69] The Kubernetes Authors. *Kubernetes Documentation*. URL: <https://kubernetes.io/docs/> (visited on 10/10/2026).
- [70] *namespaces(7) — Linux manual page*. URL: <https://man7.org/linux/man-pages/man7/namespaces.7.html> (visited on 10/10/2026).
- [71] *cgroups(7) — Linux manual page*. URL: <https://man7.org/linux/man-pages/man7/cgroups.7.html> (visited on 10/10/2026).
- [72] Brendan Burns et al. “Borg, Omega, and Kubernetes”. In: *Communications of the ACM* 59.5 (2016), pp. 50–57. DOI: 10.1145/2890784.
- [73] Falk Bartels et al. *Summary of the Run 3 in situ Phase-II NSW + FELIX + Readout + Dataflow Test*. Tech. rep. Geneva: CERN, 2026. URL: <https://cds.cern.ch/record/2970613>.

- [74] The CMS Collaboration. “Search for Narrow Resonances in Dijet Final States at $\sqrt{s} = 8$ TeV with the Novel CMS Technique of Data Scouting”. In: *Phys. Rev. Lett.* 117 (3 July 2016), p. 031802. DOI: 10.1103/PhysRevLett.117.031802. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.117.031802>.
- [75] A. Puig. “The LHCb Turbo stream”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 824 (2016). Frontier Detectors for Frontier Physics: Proceedings of the 13th Pisa Meeting on Advanced Detectors, pp. 38–39. ISSN: 0168-9002. DOI: <https://doi.org/10.1016/j.nima.2015.11.081>. URL: <https://www.sciencedirect.com/science/article/pii/S0168900215014552>.
- [76] The ATLAS Collaboration. “Search for Low-Mass Dijet Resonances Using Trigger-Level Jets with the ATLAS Detector in pp Collisions at $\sqrt{s} = 13$ TeV”. In: *Phys. Rev. Lett.* 121 (8 Aug. 2018), p. 081801. DOI: 10.1103/PhysRevLett.121.081801. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.121.081801>.
- [77] The ATLAS Collaboration. “Search for new resonances in mass distributions of jet pairs using 139 fb^{-1} of pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector”. In: *JHEP* 2003 (2020), p. 145. DOI: 10.1007/JHEP03(2020)145. arXiv: 1910.08447. URL: <https://cds.cern.ch/record/2694189>.
- [78] The ATLAS Collaboration. “Search for electroweak-scale dijet resonances using trigger-level analysis with the ATLAS detector in 132 fb^{-1} of pp collisions at $\sqrt{s} = 13$ TeV”. In: *Phys. Rev. D* 112 (9 Nov. 2025), p. 092015. DOI: 10.1103/15p2-bkg8. URL: <https://link.aps.org/doi/10.1103/15p2-bkg8>.
- [79] M. Bonaventura et al. *ATLAS TDAQ Phase-II Upgrade: Run 4 TLA Taskforce Interim Report*. ATLAS internal document. Version 0.2, unpublished. CERN, Dec. 2025.
- [80] The ATLAS collaboration. “Search for light resonances decaying to boosted quark pairs and produced in association with a photon or a jet in proton–proton collisions at $s=13\text{TeV}$ with the ATLAS detector”. In: *Physics Letters B* 788 (2019), pp. 316–335. ISSN: 0370-2693. DOI: <https://doi.org/10.1016/j.physletb.2018.09.062>. URL: <https://www.sciencedirect.com/science/article/pii/S037026931830830X>.
- [81] The ATLAS Collaboration. “Search for low-mass resonances decaying into two jets and produced in association with a photon or a jet at $\sqrt{s} = 13$ TeV with the ATLAS detector”. In: *Phys. Rev. D* 110 (3 Aug. 2024), p. 032002. DOI: 10.1103/PhysRevD.110.032002. URL: <https://link.aps.org/doi/10.1103/PhysRevD.110.032002>.

- [82] The ATLAS Collaboration. “Search for boosted low-mass resonances decaying into hadrons produced in association with a photon in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector”. In: *JHEP* 2501 (2025), p. 099. DOI: 10.1007/JHEP01(2025)099. arXiv: 2408.00049. URL: <https://cds.cern.ch/record/2906065>.
- [83] Daniel Abercrombie et al. “Dark Matter benchmark models for early LHC Run-2 Searches: Report of the ATLAS/CMS Dark Matter Forum”. In: *Physics of the Dark Universe* 27 (2020), p. 100371. ISSN: 2212-6864. DOI: <https://doi.org/10.1016/j.dark.2019.100371>. URL: <https://www.sciencedirect.com/science/article/pii/S2212686419301712>.
- [84] Antonio Boveia et al. “Recommendations on presenting LHC searches for missing transverse energy signals using simplified s-channel models of dark matter”. In: *Physics of the Dark Universe* 27 (2020), p. 100365. ISSN: 2212-6864. DOI: <https://doi.org/10.1016/j.dark.2019.100365>. URL: <https://www.sciencedirect.com/science/article/pii/S2212686419301633>.
- [85] The ATLAS Collaboration. *Dark matter summary plots for s-channel and dark Higgs models*. Tech. rep. All figures including auxiliary figures are available at <https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2025-041>. Geneva: CERN, 2025. URL: <https://cds.cern.ch/record/2946408>.
- [86] S. Navas et al. “Review of Particle Physics”. In: *Phys. Rev. D* 110 (3 Aug. 2024), p. 030001. DOI: 10.1103/PhysRevD.110.030001. URL: <https://link.aps.org/doi/10.1103/PhysRevD.110.030001>.
- [87] The CMS Collaboration. *Search for low-mass vector and scalar resonances decaying into a quark-antiquark pair in proton-proton collisions at $\sqrt{s} = 13$ TeV*. Tech. rep. Submitted to Physical Review Letters. All figures and tables can be found at <http://cms-results.web.cern.ch/cms-results/public-results/publications/EXO-24-007> (CMS Public Pages). Geneva: CERN, 2026. arXiv: 2603.21965. URL: <https://cds.cern.ch/record/2957809>.
- [88] Andrew J. Larkoski, Ian Moulton, and Benjamin Nachman. “Jet substructure at the Large Hadron Collider: A review of recent advances in theory and machine learning”. In: *Physics Reports* 841 (2020). Jet substructure at the Large Hadron Collider: A review of recent advances in theory and machine learning, pp. 1–63. ISSN: 0370-1573. DOI: <https://doi.org/10.1016/j.physrep.2019.11.001>. URL: <https://www.sciencedirect.com/science/article/pii/S0370157319303643>.
- [89] J. Alwall et al. “The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations”. In: *JHEP* 07 (2014), p. 079. DOI: 10.1007/JHEP07(2014)079. arXiv: 1405.0301 [hep-ph].

- [90] Christian Bierlich et al. “A comprehensive guide to the physics and usage of PYTHIA 8.3”. In: *SciPost Phys. Codebases* (2022), p. 8. DOI: 10.21468/SciPostPhysCodeb.8. URL: <https://scipost.org/10.21468/SciPostPhysCodeb.8>.
- [91] The ATLAS Collaboration. “Optimisation of large-radius jet reconstruction for the ATLAS detector in 13 TeV proton–proton collisions”. In: *Eur. Phys. J. C* 81.4 (2021). 68 pages in total, author list starting page 52, 21 figures, 2 tables, submitted to EPJC. All figures including auxiliary figures are available at <http://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PAPERS/JETM-2018-06/>, p. 334. DOI: 10.1140/epjc/s10052-021-09054-3. arXiv: 2009.04986. URL: <https://cds.cern.ch/record/2730141>.
- [92] UA2 Collaboration. “A Measurement of two jet decays of the W and Z bosons at the CERN $\bar{p}p$ collider”. In: *Z. Phys. C* 49 (1991), pp. 17–28. DOI: 10.1007/BF01570793.
- [93] Glen Cowan et al. “Asymptotic formulae for likelihood-based tests of new physics”. In: *Eur. Phys. J. C* 71 (2011). [Erratum: *Eur.Phys.J.C* 73, 2501 (2013)], p. 1554. DOI: 10.1140/epjc/s10052-011-1554-0. arXiv: 1007.1727 [physics.data-an].