High-performance storage and dataflow solutions for the data acquisition system of particle physics experiments

Adam Abed Abud

UNIVERSITY OF LIVERPOOL

A thesis presented for the degree of

Doctor of Philosophy

July 2022

Department of Physics
Oliver Lodge Laboratory
University of Liverpool

إذا غامَرْتَ فيشَرَفٍ مَرُومِ
فَلا تَقنَعْ بما دونَ النّجومِ

أبو الطيب المتنبي

Acknowledgements

This thesis work would not have been possible without the supervision of many talented people to whom I will be forever grateful. First and foremost, I want to thank Karol Hennessy for the guidance and directions as my doctoral advisor. I also want to thank Giovanna Lehmann Miotto as my CERN supervisor for the patient support and constant supervision over the past years. Working closely with both of you was extremely rewarding. Thank you.

I am grateful to Christos Touramanis and the University of Liverpool for the opportunity to pursue this doctorate.

I owe a lot to Fabrice Le Goff for the feedback over many years of working closely together in ATLAS.

I also want to thank Leigh Whitehead for the help on the DUNE analysis. Thanks to Marco Roda for the many ideas, suggestions, and discussions. Let me also thank Matias Bonaventura, working with you on the ATLAS DAQ was a great pleasure.

I also have to thank Andrea Negri, who first introduced me to CERN and the field of data acquisition. Thank you for everything, I will be forever grateful.

I had the privilege of working with many talented people in both the ATLAS and DUNE collaborations. Thank you for making this journey memorable.

Special thanks to all my friends. If I managed to get here is also thanks to all of you.

Finally, I would like to thank my family for their unconditional support, patience and love.

Declaration

I declare that this thesis is the result of my own work, except where a reference is made to the work of others. This thesis has not been submitted, in whole or in part, for another qualification to this, or any other, university.

Adam Abed Abud

Abstract

The data acquisition system of particle physics experiments is a mission-critical component responsible for the experiments' success. The next generation of large-scale particle physics experiments will have millions of sensors producing a large amount of data at high rates. The DUNE and Phase-II ATLAS experiments are expected to start taking data in the late 2020s. Data rates from both detectors will reach orders of terabytes per second, posing a significant challenge to the data acquisition chain and, especially, to the storage and dataflow system which will need to be designed accordingly. Therefore, it becomes essential to investigate methods to collect, store and transport the data efficiently. This thesis presents the work done on the performance characterization of different storage technologies and dataflow methods that are suitable for implementing the storage systems of both the ATLAS and DUNE experiments.

Persistent memory devices and novel solid-state devices have been investigated as a potential solution to implement the local storage system of the DUNE experiment. This was achieved using both synthetic benchmarks with an emulated workload and integration testing for each storage technology. Performance results obtained after carefully tuning the devices show that such technologies can sustain the target rates required by the experiment.

A distributed high-throughput key-value store (DAQDB) was designed for the data acquisition task and it was extensively tested as a solution for the large storage buffer of the ATLAS experiment. The results show that the current implementation of DAQDB cannot sustain the target bandwidth required by the ATLAS experiment. Therefore, the work was followed by the investigation of dataflow methods that combine local storage management solutions as a possible means to achieve the target goals of the experiment. An extensive study on the evolution of the storage system with discrete event simulations was also done to understand the advantages and limitations of different data acquisition architectures.

The experimental research was also completed by investigating a novel algorithm (SparseNet) designed to classify track and shower energy deposits across a liquid argon detector. The SparseNet algorithm was extensively tested with both Monte Carlo data and beam data from a test setup of the DUNE experiment available at CERN. Preliminary results show that the SparseNet outperforms the currently adopted track and shower classification algorithm.

Table of Contents

List of Abbreviations

DAQ	Data AcQuisition
DF	Dataflow system
DH	Data Handler system
DUNE	Deep Underground Neutrino Experiment
EF	Event Filter
EM	Electro-magnetic
KVS	Key-value store
MC	Monte Carlo
PD	Photon Detector
RU	Readout Unit
SH	Storage Handler system
SNB	SuperNova Burst event
SSD	Solid-state device
SURF	Sanford Underground Research Facility
TDAQ	Trigger and Data Acquisition
TPC	Time Projection Chamber

1

Introduction

Large-scale modern particle physics experiments consist of millions of sensors, producing a substantial amount of data at high rates. The ATLAS detector [1] at CERN and the planned DUNE Far Detector [2] at the Sanford Underground Research Facility are two examples of such experiments. In particle physics, the data acquisition (DAQ) is the system that is mainly responsible for collecting and processing the signals produced by the various detectors (readout), selecting the most interesting information (data selection) and transferring and saving the data on permanent storage (storage and dataflow system). The DAQ system is essential for the operation of a particle physics experiment.

The planned upgrade of the ATLAS experiment and the start of the DUNE experiment are both expected to take place in the late 2020s. Data rates from the detectors will reach orders of terabytes per second, posing a significant challenge to the storage and dataflow system of the data acquisition of the experiments. Therefore, such systems will need to be adapted to accommodate the specifics of the various tasks: buffer, format, transport or store requested data. This is done by combining local storage solutions and dataflow policies to orchestrate the flow of data effectively.

In the case of the ATLAS experiment, a large persistent storage buffer is planned to decouple the data processed by the readout system from the data selection. At high data rates, an intermediate high-performance storage system becomes particularly useful to dimension the data processing part of the system for an average load without needing to sustain temporary peaks. In the case of the DUNE experiment, one of the physics goals is to detect and store neutrino interactions from core-collapsing supernova events. In this case, once the event has been detected, a high-throughput data path is activated from the readout nodes to the storage system.

Although both the ATLAS and DUNE experiments have similar incoming data rates, similar size for the data acquisition infrastructure and both plan to use high-performance storage solutions, the specific technologies have to be carefully tuned for the task within the DAQ system in order to best support the data taking needs of the experiments.

1.1 Motivation for this work

Emerging high-performance storage technologies are being used to design new architectures where a large storage buffer is used to decouple data production and processing. This is motivated by the trend in industry where high-performance computing environments are experiencing a considerable increase in data volumes, making data consumption more difficult. Similarly, in current data acquisition systems, high I/O rates are expected between data-producing nodes (readout) and data-consuming nodes (data selection). Therefore, it becomes crucial to investigate the use of scalable and high-throughput storage technologies in the context of the next-generation DAQ system of particle physics experiments.

The storage system of a particle physics experiment is an essential component for the whole experiment. The data from the detectors need to be safely archived in persistent storage media in order to support the wide physics program of the experiment. In order to achieve this goal, the storage system needs to cope with the high throughput and high data volume from the front-end electronics. In this research work, several storage devices and software solutions have been investigated (benchmarking, testing, and tuning) to evaluate their performance and feasibility as local buffering solutions, e.g. local storage system for the DUNE supernova buffer.

1.2 Structure of the document

This thesis is organized into the following chapters:

- Chapter 2, *Introduction to Neutrino Physics*, reviews the neutrino physics with a focus on the neutrino oscillations.

- Chapter 3, *The DUNE experiment*, gives an overview of the DUNE experiment from the physics objectives to the description of the detector components.

- Chapter 4, *Particle identification in LAr detectors*, proposes the use of an innovative algorithm based on Deep Learning for the classification between track and shower energy deposits for the DUNE experiment.

- Chapter 5, *Data acquisition system of the DUNE experiment*, presents the DUNE trigger and data acquisition system.

- Chapter 6, *ATLAS TDAQ system for the Phase-II upgrade*, presents a description of the ATLAS data acquisition system for the Phase-II upgrade.

- Chapter 7, *High-performance storage buffer for supernova events*, analyzes different storage device technologies as a possible implementation for the DUNE local storage buffer. The results from the synthetic benchmarks, testing with emulators and integration with the DAQ are presented in detail.

- Chapter 8, *Dataflow methods in particle physics experiments*, proposes different solutions to build a scalable, high-throughput storage system. Testing and integration is done in the context of the ATLAS experiment.

- Chapter 9, *Conclusion and future work*, summarizes the main achievements of this research work, including the list of contributions and future work activities.

1.3 Contributions

The author has contributed in several areas of the data acquisition system of particle physics experiments. The novelty of this research work lies in investigating, benchmarking and integrating modern storage and dataflow solutions for the data acquisition system of both the ATLAS and DUNE detectors. The evaluation of different storage technologies has been done with synthetic benchmarks and emulated workloads. This was followed by an in-depth evaluation of the feasibility of such technologies with the data acquisition system. The physics objectives of the experiment (e.g. storage buffer for supernova events) have always been prioritized; therefore, the integration testing with the whole DAQ system and with the detector apparatus was favored instead of relying solely on synthetic benchmarks.

In the area of dataflow, the author has contributed to the development of the DAQ of both the ATLAS and DUNE collaborations. A novel key-value object store based on cutting-edge hardware platforms was extensively tested and later integrated with the ATLAS DAQ system. This investigation aimed to understand the scalability, throughput performance and operation of such a commercial off-the-shelf solution. In parallel, the author has also worked on developing local storage solutions and dataflow strategies as a possible implementation of the storage system of the ATLAS experiment. This was followed by an evaluation of the feasibility and cost-benefit of the overall storage system as well as simulation studies of different features of the ATLAS Dataflow system.

As part of the experimental research, an important activity of this work was also dedicated to understanding the DUNE detector and, in particular, distinguishing between track and shower energy deposits. These represent an essential signature for many physics signals. A new algorithm based on Deep Learning has been evaluated in detail and tested in a prototype of the DUNE experiment located at CERN.

2

Introduction to Neutrino Physics

The objective of this chapter is to provide an introduction to neutrino physics. In particular, the focus is on neutrino oscillations because they are the main part of the DUNE physics program.

2.1 Neutrino particles

The Standard Model (SM) of particle physics describes neutrinos as elementary left-handed particles electrically neutral and with zero mass. Neutrinos interact with the weak force; they are mostly unaffected by the gravity interaction and do not participate in the strong interaction. Therefore, neutrinos typically pass through matter without being detected and without interacting, making them a very challenging elementary particle to investigate.

Extensive research is ongoing to further understand the properties of neutrinos and the shortcomings of their description in the SM:

- the mass value of the three neutrino flavors;
- evidence of lepton number violation such as in neutrino-less double beta decay
- evidence for charge-parity (CP) violation by studying the oscillations of neutrinos and anti-neutrinos

2.2 Neutrinos in the Standard Model

According to the Standard Model, neutrinos are massless particles with no electric or color charges. Neutrino particles are divided into three flavors states (also known as neutrino generations) that only undergo weak interactions and that can be observed experimentally: electron neutrino (ν_e), muon neutrino (ν_μ) and tau neutrino (ν_τ). The neutrino flavor is conserved due to the massless nature of the particles.

Neutrinos interact with other particles via the exchange of Z^0 or $W^\pm$ bosons. Interactions with hadrons can be classified according to the boson exchanged between the lepton current and the hadronic current. In charged-current (CC) interactions the exchange boson is a $W^\pm$ boson and neutral current (NC) interactions occurs with the exchange of a Z^0 boson.

Experimentally the neutrino flavor can be identified only in CC interactions by tagging the flavor of the final state charged lepton. Specifically, the interaction

$$\nu_e + Ar \longrightarrow e^- + Ar + \pi^+$$

is an example of CC interaction where in the final state an electron is observed and it allows to determine the flavor of the incoming neutrino.

2.3 Neutrino oscillations

Neutrinos are also classified into three mass eigenstates (ν_1, ν_2, ν_3) which determine how neutrinos propagate in time and in space. One of the properties of neutrinos is that they change between flavor states in what is known as the *neutrino oscillation* phenomenon. This arises since the weak interaction eigenstates (ν_e, ν_μ, ν_τ) are not the same as the mass eigenstates (ν_1, ν_2, ν_3). The coefficients of the linear combination of the neutrino states define a 3×3 matrix (U) known as the *neutrino mixing matrix*, *leptonic mixing matrix* or Pontecorvo-Maki-Nakagawa-Sakata (PMNS) matrix [3].

Therefore, the wavefunction of a neutrino flavor can be expressed as a liner combination of mass eigenstates:

$$|\nu_\alpha\rangle = \sum_{i=1}^{3} U^*_{\alpha i}\,|\nu_i\rangle \tag{2.1}$$

where α can be any of the three neutrino flavors and $U^*_{\alpha i}$ are the complex conjugates of the PMNS matrix elements. Similarly, the mass eigenstates can be expressed as a linear combination of the flavor eigenstates:

$$|\nu_k\rangle = \sum_\alpha U_{\alpha k} |\nu_k\rangle \tag{2.2}$$

The PMNS matrix U can be decomposed into a product of 3x3 matrices as illustrated in equation 2.3:

$$\begin{aligned} U_{PMNS} &= \begin{pmatrix} U_{e1} & U_{e2} & U_{e3} \\ U_{\mu 1} & U_{\mu 2} & U_{\mu 3} \\ U_{\tau 1} & U_{\tau 2} & U_{\tau 3} \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_{23} & s_{23} \\ 0 & -s_{23} & c_{23} \end{pmatrix} \begin{pmatrix} c_{13} & 0 & s_{13}e^{-i\delta_{\mathrm{CP}}} \\ 0 & 1 & 0 \\ -s_{13}e^{i\delta_{\mathrm{CP}}} & 0 & c_{13} \end{pmatrix} \begin{pmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{aligned} \tag{2.3}$$

where $s_{ij} = \sin\theta_{ij}$ and $c_{ij} = cos\theta_{ij}$. The PMNS matrix is parameterized by three real mixing angles ($\theta_{12}, \theta_{23}, \theta_{13}$) and a single CP-violating complex phase ($e^{i\delta}$). Therefore, the matrix in equation 2.1 can be written as:

$$U^*_{\alpha k} = \begin{pmatrix} U^*_{e1} & U^*_{e2} & U^*_{e3} \\ U^*_{\mu 1} & U^*_{\mu 2} & U^*_{\mu 3} \\ U^*_{\tau 1} & U^*_{\tau 2} & U^*_{\tau 3} \end{pmatrix} \tag{2.4}$$

The evolution of the neutrino wavefunction is obtained by applying the time evolution operator to the mass eigenstates. Note that each of the flavor states evolves with a different phase $\phi_i(\mathbf{x}, t) = p_i \cdot x = (E_i t - \mathbf{p_i} \cdot \mathbf{x})$. Therefore,

$$|\nu(\mathbf{x}, t)\rangle = \sum_i U^*_{\alpha i} e^{-i\phi_i(\mathbf{x},t)} |\nu_i\rangle \tag{2.5}$$

The probability of oscillation between two neutrino flavors can then be calculated as:

$$\begin{aligned} P(\nu_\alpha \to \nu_\beta) &= |\langle \nu_\beta | \nu(\mathbf{x},t)\rangle|^2 \\ &= |\sum_i^3 U^*_{\alpha i} U_{\beta i} e^{-i\phi_i}|^2 \\ &= \sum_i^3 \sum_j^3 U^*_{\alpha i} U_{\beta i} U_{\alpha j} U^*_{\beta j} e^{-i(\phi_i - \phi_j)} \end{aligned} \tag{2.6}$$

By taking into account the ultra-relativist limit, the mass of the neutrinos becomes negligible compared to the momentum. Thus, it is possible to translate the $(\phi_i - \phi_k)$ terms into squared mass differences which are the parameters used for the calculation of the neutrino oscillation probabilities. In this context,

$$m_i \ll \mathbf{p}$$

$$\mathbf{x} \simeq ct$$

Thus, by using the Taylor expansion series and by using the fact that $\lim_{m\to+0} E = \mathbf{p}$ (in natural units):

$$E_i = \sqrt{|\mathbf{p}|^2 + m_i^2} = \mathbf{p}\sqrt{1 + \frac{m_i^2}{|\mathbf{p}|^2}} \simeq \mathbf{p} + \frac{m_i^2}{2\mathbf{p}} \tag{2.7}$$

$$\phi_i(\mathbf{x},t) \simeq \frac{m_i^2}{2p} x \tag{2.8}$$

Therefore, equation 2.6 can be rewritten as:

$$P(\nu_\alpha \to \nu_\beta) = \sum_{i,j=1}^3 U^*_{\alpha i} U_{\beta i} U_{\alpha j} U^*_{\beta j} \exp(-i\frac{\Delta_{ij}}{2p} x) \tag{2.9}$$

where Δ_{ij} depends on the mass squared difference $\Delta m_{ij}^2 = m_i^2 - m_j^2$. Let L be the total distance that has been travelled and E_ν the neutrino energy:

$$P(\nu_\alpha \to \nu_\beta) = \sum_{i,j=1}^3 U^*_{\alpha i} U_{\beta i} U_{\alpha j} U^*_{\beta j} \exp(-i\frac{\Delta_{ij}}{2E} L) \tag{2.10}$$

Equation 2.10 represents the probability of oscillation a neutrino with flavor α to flavor β. By parametrizing the oscillation probability in terms of mass splittings, by using the mixing angles and the real and imaginary components of the PMNS matrix we obtain the following:

$$P(\nu_\alpha \to \nu_\beta) = \delta_{\alpha\beta} - 4\sum_{i>j} Re(U^*_{\alpha i}U_{\beta i}U_{\alpha j}U^*_{\beta j}) \sin^2(\frac{\Delta^2_{ij}L}{4E})+ \\ + 2\sum_{i>j} Im(U^*_{\alpha i}U_{\beta i}U_{\alpha j}U^*_{\beta j}) \sin(\frac{\Delta^2_{ij}L}{2E}) \quad (2.11)$$

where the first term ($\delta_{\alpha\beta}$) corresponds to the case in which there are no neutrino oscillations. The middle term represents the flavor oscillations and it is given by the real components of the PMNS matrix which define the amplitude of the oscillation and the mass splitting component defines the phase of the oscillation. Finally, the last term takes into consideration the impact of CP-violation which is the violation of both the charge and parity symmetries. The resulting probability is used by neutrino experiments to determine the expected rates of oscillations for neutrinos in vacuum.

A very interesting property that arises from the oscillation probability of two neutrino flavors is that if the phases were all the same then there would be no oscillation. Flavour oscillation is, therefore, possible only with massive neutrinos. Although neutrinos are described as massless particles in the SM, neutrino flavor oscillation measurements, like in the T2K experiment [4], have given direct evidence that the mass difference between two flavors of neutrinos is not zero [4].

Note that from equation 2.10 the neutrino oscillation becomes appreciable when $\frac{\Delta ijL}{2E}$ is $\mathcal{O}(1)$. Therefore, an oscillation experiment becomes sensitive to a mass difference between neutrino flavors when the following condition between the neutrino energy E_ν and the source-detector distance L is satisfied:

$$\Delta m^2 \sim \frac{E[GeV]}{L[km]} \quad (2.12)$$

Therefore, the design of neutrino experiments is done by carefully selecting the energy of the incoming neutrinos and the distance where the detector apparatus is placed. Table 2.1 shows the characteristic values of L and E for different neutrino sources and for both Short Baseline (SBL) and Long Baseline (LBL) neutrino experiments. The corresponding mass splitting $|\Delta m^2|$ is also included.

Experiment	**L (m)**	**E (MeV)**	$\|\Delta m^2\|$ **(e**V^2**)**
Atmospheric	$10^4 - 10^7$	$10^2 - 10^5$	$10^{-1} - 10^{-4}$
Reactor	SBL: $10^2 - 10^3$ LBL: $10^4 - 10^5$	1	$10^{-2} - 10^{-3}$ $10^{-4} - 10^{-5}$
Accelerator	SBL: 10^2 LBL: $10^5 - 10^6$	$10^3 - 10^4$ $10^4 - 10^5$	≥ 0.1 $10^{-2} - 10^{-3}$

Table 2.1: Characteristic values of L and E for different neutrino sources and for both Short Baseline (SBL) and Long Baseline (LBL) neutrino experiments [5].

Finally, note that in equation 2.11 the Δ^2_{ij} values and the mixing angles θ are known parameters, therefore by measuring the oscillation probabilities between neutrino flavours it is possible to evaluate, for example, the CP-violation term. It is also worth mentioning that with neutrino oscillation experiments only the differences of the squared neutrino masses (Δm^2_{ij}) can be studied. Absolute mass measurements can be found, for instance, in nuclear experiments by investigating the electron energy distribution of the β-decay of tritium (KATRIN experiment [6]).

2.4 Overview of the DUNE oscillation analysis

At its core, an oscillation analysis is a counting experiment in which neutrino interactions are divided according to the flavor of the neutrino produced in the interaction and the total energy released in the interaction. These samples are collected in two different sites (near and far) along the neutrino beam and a comparison between the spectra between the different sites allows the extraction of the oscillation parameters. There are two types of oscillation analyses: appearance and disappearance. A disappearance analysis focuses only on a single neutrino flavor and it aims at measuring the reduced rate for that flavor in the far site because of the oscillation into another neutrino flavor that may not be detected by the far site. The disappearance analysis corresponds to the situation when $\alpha = \beta$ in equation 2.11. On the other hand, an appearance analysis focuses on the detection of a flavor in the Far Detector that is different from the initial one. In order to measure the CP-violation, an appearance experiment is necessary since the oscillation probability (equation 2.11) is independent on the CP-phase if the two neutrino flavors are the same.

DUNE aims to measure the CP-violation mainly using the ν_μ to ν_e channel. In order to achieve this it is necessary to have the capability to distinguish electrons and muons from the final state as well as to separate them from the hadronic background of the interaction. In addition to the particle identification requirement, it is necessary to be able to reconstruct the energy released in the interaction.

In the DUNE experiment, the physics requirements for the neutrino oscillations analysis are achieved by having a detector that is capable of acting both as a tracker and as a calorimeter. Tracking is necessary for separating tracks originating from electrons and muons, whereas a calorimeter is necessary for reconstructing the energy released in the interactions. A liquid argon time projection chamber (LArTPC) is the ideal candidate for achieving the targeted objectives of the DUNE experiment.

3

The DUNE experiment

3.1 Introduction

As discussed in the previous chapter, many questions in neutrino physics are still unanswered. The Deep Underground Neutrino Experiment (DUNE) is an international experiment dedicated to neutrino science [3], and that is expected to start taking data by the end of this decade. The objective of DUNE is to study mainly neutrinos originating from both accelerators and supernova events. Two detectors are planned for DUNE: a Near Detector (DUNE-ND) and a Far Detector (DUNE-FD). The former will be placed near the source of a high-intensity neutrino beam at the Fermi National Accelerator Laboratory (FNAL) in Batavia (Illinois). The neutrino beam source originates from the interaction of high-energy protons with a target material. The PIP-II accelerator complex is a 215 m long machine that is responsible for accelerating protons to an instantaneous power of approximately 1 MW. In contrast, the DUNE-FD will be installed at the Sanford Underground Research Laboratory (SURF) in South Dakota more than 1300 km from the neutrino source. This chapter provides a general introduction to the DUNE physics program, a description of the DUNE Far Detector experimental setup, and an overview of a prototype setup available at CERN (ProtoDUNE-SP experiment).

3.2 DUNE's Physics objectives

The primary objectives of the DUNE physics program can be summarized in three categories:

- High sensitivity measurements of neutrino oscillation parameters
- Physics beyond the Standard Model (BSM)
- Neutrinos from core-collapse supernova events in the Milky Way galaxy

3.2.1 Neutrino oscillation

One of the primary goals of the DUNE physics program is the measurement of the oscillation of ν_μ and $\overline{\nu}_\mu$ in a range of energies between 0.1 MeV and 10 MeV [3]. In fact, the 1300 km DUNE baseline design offers the possibility to study the potentially large asymmetry in the oscillation between $\nu_\mu \rightarrow \nu_e$ and $\overline{\nu}_\mu \rightarrow \overline{\nu}_e$. This is further highlighted in figure 3.1 which shows the appearance probability of the $\nu_\mu \rightarrow \nu_e$ and $\overline{\nu}_\mu \rightarrow \overline{\nu}_e$ as a function of the neutrino energy assuming the detector is located 1300 km away from the source. As can be noted, the value of the phase δ_{CP} of the PMNS matrix (equation 2.11) has an impact on both the amplitude and the phase of the neutrino oscillation. In addition, from figure 3.1 it can also be noted that the amplitude of the oscillation probability is higher for neutrino energies less than 1.5 GeV.

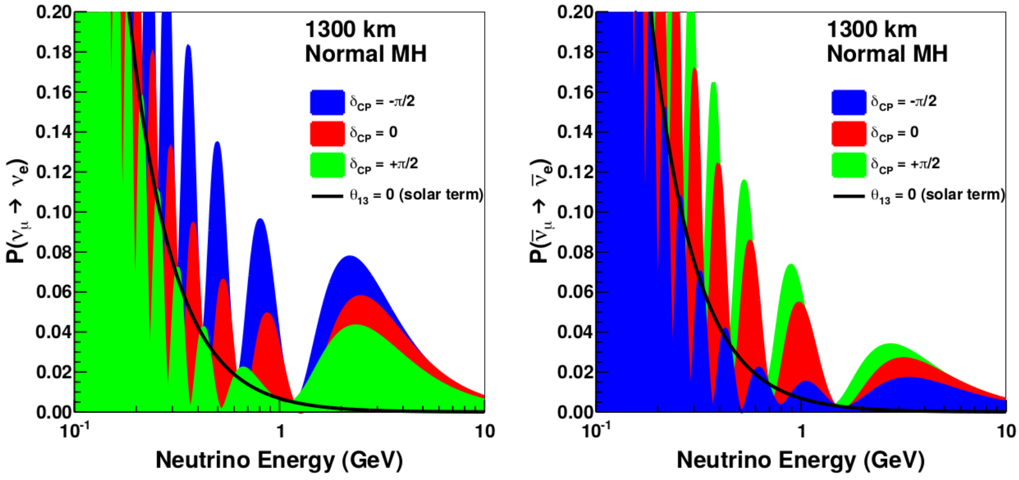


Figure 3.1: Appearance probability of $\nu_\mu \rightarrow \nu_e$ and $\overline{\nu}_\mu \rightarrow \overline{\nu}_e$ as a function of the neutrino energy and phase δ_{CP} for both neutrinos (left) and antineutrinos (right). Figure taken from [3].

3.2.2 Physics Beyond the Standard Model (BSM)

As an additional physics research line, the DUNE experiment will also be dedicated to investigating the proton decay and other baryon-number violating process [3] which are only possible with a physics description beyond the Standard Model. This is possible thanks to the large liquid argon active volume and the high-intensity neutrino beam. In the case of the proton decay, the experimental signature that DUNE is looking for is given by the process $p \to K^+ + \overline{\nu}$. Current estimates suggest that it will be possible to improve the lower limits of the proton lifetime. In fact, DUNE sensitivity studies indicate that with a 30$\%$ detection efficiency the 90$\%$ CL lower limit estimate for the proton lifetime is 1.3×10^{34} years (assuming a 400-kt year exposure) [3].

Other BSM processes that may be relevant for the DUNE detector consist in investigating sterile neutrinos[1] with the ν_τ appearance or studies on the Dark Matter annihilation from the core of the sun: typically resulting in both a low and high energy neutrino component. More details on BSM studies with the DUNE detector can be found in [3].

3.2.3 Core-collapse Supernova

Another objective of the DUNE detector will be the study of the ν_e flux originating from a core-collapse supernova [3]. As confirmed by the SN1987a supernova observation, neutrinos from core-collapse events can reveal rich information about the structure and properties of exploding stars which are not typically available from electro-magnetic (EM) signatures. A massive star will lose energy by neutrino emission originating from the pair annihilation process during the end of its life. This event happens when the temperature of the star reaches T $\sim$ 10^{10} K and the density $\rho \sim 10^{10}\ g/cm^3$. Therefore, the star's iron core cannot sustain the equilibrium condition because nuclear fusion is no longer favored. As a result, when the mass of the star reaches the critical value of $1.4\ M_\odot$, it will collapse, releasing 10^{53} ergs and over 10^{58} neutrinos with an energy spectrum peaked around 10 MeV [7].

The detection principle of supernovae neutrinos is based on the interaction with argon nuclei, outlined in equation 3.1:

$$\nu_e + {}^{40}Ar \longrightarrow e^- + {}^{40}K^* \tag{3.1}$$

More details on the detection of supernova neutrino events is provided in chapter 7.

[1] Sterile neutrinos are a hypothesized class of neutrinos that interact with only the gravitational force.s

3.3 Overview of a liquid argon TPC detector

The DUNE experiment is based on a LArTPC (liquid Argon time projection chamber) detector whose detection principle is to collect the charge produced by ionization electrons and scintillation light from charged particles passing the liquid argon volume. The argon in the TPC acts both as a detection medium and a target from which the photons and charged particles are produced and detected. The operating principle of a LArTPC based on the horizontal drift technology, illustrated also in figure 3.2, consists of liquid argon placed between an alternating set of wires (anodes) and wire mesh (cathode). Each pair of anode and cathode generate a uniform electric field along which ionization electrons drift: electrons move from the cathode to the anode wires. Ultimately, the charge of the electrons produced by the drifting ionization process is collected to process the signal. The wire configuration of the DUNE-FD LArTPC is to have three planes of wires to measure the ionization: two induction planes (U and V planes) and one collection plane (Y plane). Bipolar signals are formed on the induction planes as the electrons drift first towards the two wires and then away from the wires as they are collected on the collection plane. By positioning the induction plane wires in two different directions with respect to the collection plane, it is then possible to obtain a 2D projection of the charged particle position as it traverses the liquid argon volume. The third spatial dimension can be calculated by measuring the time for the ionizing electron to finish drifting. In fact, the drift velocity of electrons in a given electric field is constant, and thus:

$$x = t_{drift} * v_{drift} = (t_f - t_0) * v_{drift} \tag{3.2}$$

where x is the third coordinate, t_f is the final time after the electron finished drifting, t_0 is the time at the start of the charge drift and v_{drift} is the known drift velocity in liquid argon.

Figure 3.2 illustrates the signal formation process in a LArTPC detector. In particular, it depicts a neutrino interaction that produces two charged particles that ionize the liquid argon and produce electrons that drift towards the anode, in the direction opposed to the electric field. An overview of the V induction plane and the Y collection plane is also shown in the figure.

Furthermore, charged particles produced because of the neutrino interactions lose energy by ionizing the liquid argon medium of the TPC. Therefore, measuring the amount of ionization makes it possible to estimate the energy lost by the charged particles as they traverse the detec-

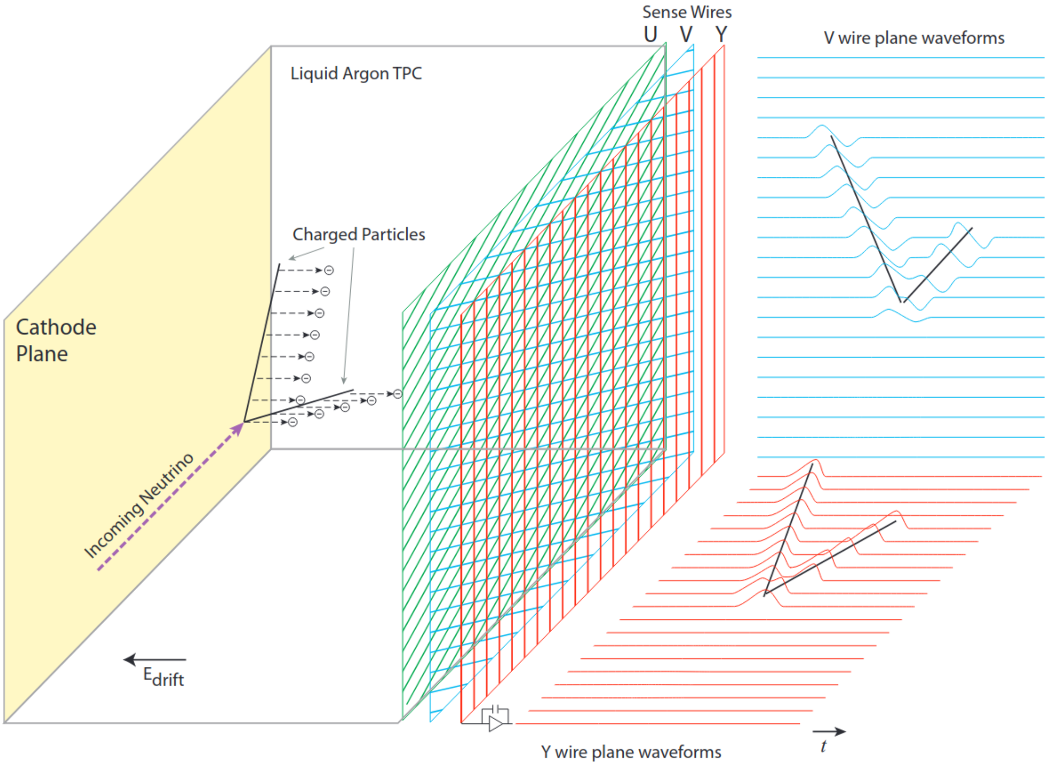


Figure 3.2: Illustration of signal formation in a LArTPC detector. Overview of the V induction plane and the Y collection plane. Figure taken from [8].

tion volume. This is done by measuring the size of the signals produced on the wires. By doing so, LArTPC detectors offer great discriminating power between electron and photon showers, which is important in accelerator neutrino experiments where photons are often background signals for electron neutrino interactions. In fact, photons that convert through pair production into an electron and positron pair can be distinguished from single electrons by measuring the energy deposited at the beginning of the shower. An electron-positron pair usually deposits twice the energy lost by a single electron through the ionization process in the liquid argon.

In addition to ionization, another process that is produced by charged particles in LArTPC is the scintillation light. This is produced by the excitation of argon nuclei and their subsequent decay. Scintillation light in liquid argon is produced isotropically at around 128 nm and it is divided into two components: slow and fast. The former has a decay time of approximately 5 ns, whereas the latter has a decay time of approximately 1.3 μs. Note that liquid argon is transparent to its scintillation light. Therefore, photon detectors placed at the edges of the apparatus can be used to detect the photons produced by the charged particles. From an experimental point of view, scintillation light is typically used for timing and triggering in the operation of neutrino experiments.

Signal degradation

One crucial effect that occurs in LArTPC detectors is the recombination of ionized electrons with Ar nuclei and the subsequent production of positive Ar^+ ions. The result of this process is that part of the energy induced by the charged particle is not being collected as the electron thermalizes with the argon nuclei. Note that this is usually taken into account in the calibration of the detector and by using theoretical models to treat the recombined electrons.

Ionization attenuation refers to the process to which some of the ionization electrons are captured by impurities in the liquid argon during the drift towards the anode plane. In fact, impurities in the LAr such as water and oxygen (O_2) can reduce the total collected charge. The ionization attenuation is modelled with an exponential decay as detailed in equation 3.3:

$$Q_C = Q_0 e^{-t_d/\tau} \tag{3.3}$$

where Q_C is the collected charge, Q_0 is the initial charge deposited, t_d is the drift time and τ is the electron drift time which takes into account the presence of impurities in the LAr. In the case of the DUNE experiment, it is expected to have a contamination of the argon no greater than 100 ppt O_2-equivalent and 25 ppm N_2 [9]. This is done to ensure an electron lifetime that is sufficient for the length of the detector and to have enough light yield.

Space Charge Effect

As a charged particle traverses the LAr, the ionized electrons drift towards the anode and the positively charged argon Ar^+ ions drift towards the cathode. However, the drift velocity of the positive ions is much smaller than the electron drift velocity. The consequence of such an imbalance results in an accumulated positive charge that can distort the electric field. This phenomenon is known as Space Charge Effect (SCE) and it is especially relevant for LArTPC detectors operated on the surface. This is due to the significant rate of cosmic mouns that leads to a non-negligible accumulation of positive charge. In addition, by altering the electric field the SCE leads to both a distortion in the reconstructed trajectories of the charged particles and the energy reconstruction measurements. Therefore, when operating a LArTPC detector is highly crucial to take into account the SCE.

3.4 The DUNE Far Detector

The DUNE experiment relies on two detectors: the Near Detector (DUNE-ND) and the Far Detector (DUNE-FD). The DUNE-ND provides constraints on the neutrino flux and measures neutrino cross sections at the beam source, whereas the DUNE-FD is the apparatus mainly responsible for measuring the neutrinos after traversing the oscillation distance. At the time of writing, the design of the DUNE-ND detector has been finalized, although some implementation aspects still need to be confirmed. This research work has been focused only on the DUNE-FD. Therefore, the remaining sections of this chapter will be dedicated only on the description of the DUNE-FD.

The DUNE-FD will consist of four liquid argon TPC (LArTPC) modules placed more than a kilometer underground. Many TPC technologies have been evaluated for the DUNE-FD modules. The planned first two modules will be placed in liquid argon and they will operate with a horizontal and vertical drift. The technology of the two remaining modules still needs to be confirmed at the time of writing. Nonetheless, the data acquisition system of all the modules is going to be the same. This thesis work was done in the context of the DUNE-FD first module based on the horizontal drift TPC technology which is going to be described in detail in the next sections.

Assembly and installation of the first DUNE-FD module are expected for 2024 and 2025. This ensures that the detectors are ready to take data by 2028 as a long integration and commissioning process is needed to test the detectors and the electronics. Figure 3.3 shows an illustration of the experimental cavern at SURF for the Far Detector. The large areas marked in red represent the location of the first two modules and the central area in between represents a smaller cavern dedicated for the data acquisition and the cryogenics.

Horizontal Drift

A DUNE-FD module based on the Horizontal Drift LArTPC technology (also known historically as Single Phase or SP) will consist of a total mass of 17.5 kton, with a fiducial mass of 10 kton for the active volume. As mentioned in section 3.3, the first DUNE-FD LArTPC detector module is divided into an alternating set of five cathode and anode wire walls inside which particles can drift. The total TPC size is 12 m (height), 14 m (width) and 58.2 m (longitudinal). The dimensions of the detector providing a 3.5 m drift length. Each cathode wall

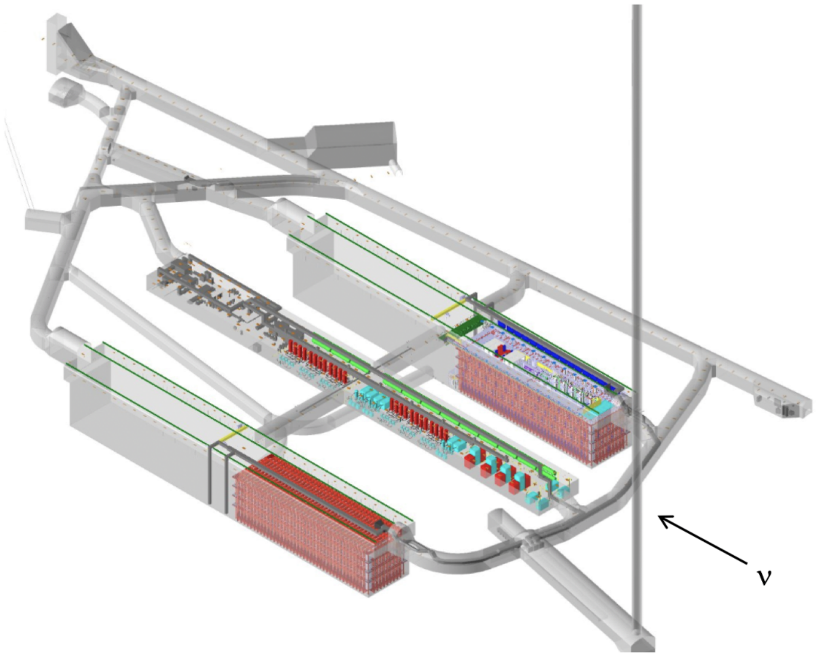


Figure 3.3: Illustration of the experimental cavern at SURF. The two large areas in red represent the location of the two detector modules. The central area represents the service cavern for the data acquisition and cryogenics. Figure taken from [9].

is called cathode plane assembly (CPA) array and it is made by three rows of 50 CPA units of size 1.2 m and 4 m. On the other hand, the anode walls are made of two rows of 25 Anode Plane Assemblies (APAs) for a total number of 150 units of 6 m and 2.3 m in size. Figure 3.4 shows the schematic representation of a DUNE-FD Horizontal Drift module, highlighting the alternating sets of APA walls (3 in total) and CPA walls (2 in total).

The LAr used inside the TPC is cooled at a temperature of approximately 87 K and must have a very high purity. As discussed in the previous section, oxygen contamination impacts the total ionization process. Therefore, oxygen contamination in the liquid argon must be kept to a value below 100 ppt. To maintain a high purity level, the LAr is continuously cycled through a purification system.

Anode Plane Assemblies

The APAs consist of large stainless steel frames on which multiple planes of wires are installed. The signals from the wires are extracted with electronics that are either placed on the top or bottom edges of the anode walls. In addition, the wires are made from 152 μm diameter copper-beryllium (CuBe) alloy, which has been specifically selected for its high durability and yield strength. This means that the wires can sustain high stress before deformation occurs.

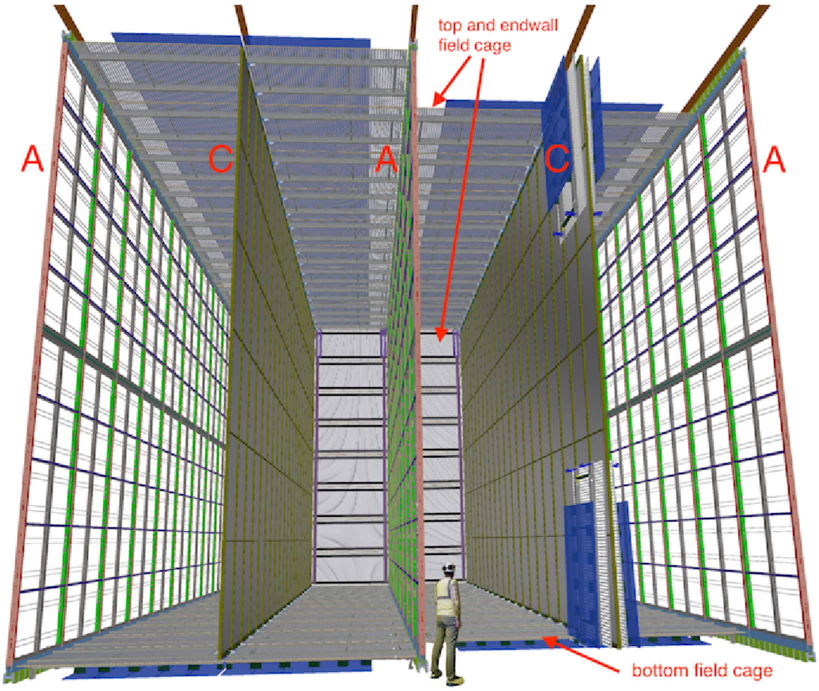


Figure 3.4: Illustration of the DUNE-FD Horizontal Drift module, showing the alternating layers of APA walls (A) and cathode walls (C). The drift areas are surrounded by the top and bottom field cages. Figure taken from [9].

Tests show that CuBe wires have a yield strength of more than 1100 MPa. Furthermore, the wire planes are assembled with an angle of $\pm 35.7^\circ$ to the vertical as shown in the schematic representation of an APA in figure 3.5 where the green and pink lines represent the two induction planes and the blue one represents the collection wires. This is to ensure that each induction wire crosses only once a collection wire.

Cathode Plane Assemblies

The successful operation of a LArTPC detector relies on the uniformity of the electric field used for the transport and collection of the ionization charge. For this reason, CPA arrays are kept at a constant voltage of -180 kV by using external power supplies to provide a nominal uniform electric field of 500 V/cm across the drift volume. In these conditions, the drift speed of electrons is 1.6 mm/μs. Note that the electric field strength is chosen to maximize the trade-off between precision in the measurements and operation of the detector. For example, the number of scintillation photons is inversely proportional to the electric field strength whereas the dE/dx measurements are affected by the electron-ion recombination effect, which decreases for high values of the electric field strength. Therefore, by setting a high value of the electric field, the energy measurements may be very accurate at the expense of having a small amount of

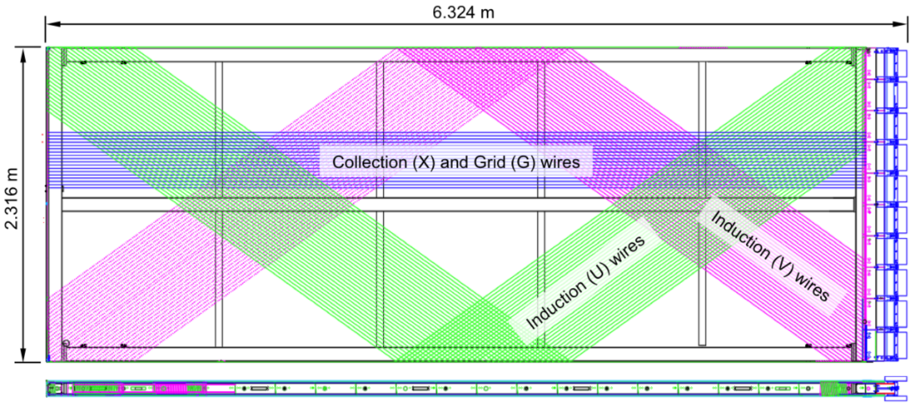


Figure 3.5: Schematic representation of an APA. Three wire planes are shown: two induction wires (U and V) and one collection plane (X). The induction wires are pictured in green and pink whereas the collection wires are in blue. The blue boxes on the right edge represent the readout electronics. Figure taken from [9].

scintillation light which, in turn, leads to an increasing difficulty in evaluating the starting time t_0 for the reconstructed particles. The nominal voltage of -180 kV and, thus, the electric field of 500 V/cm has been tuned with a trade-off between several parameters: signal to noise ratio, amount of charge collected or signal detection thresholds. This task is done by the DUNE High-Voltage system which is in charge of the operation of the detector apparatus' voltage. In addition, the tuning of the system is also based on the successful operation of other LArTPC detectors (e.g. MicroBoooNE) where an adequate signal to noise ratio was achieved [2].

3.5 The ProtoDUNE-SP experiment

The ProtoDUNE Single Phase (ProtoDUNE-SP) experiment [11] was a prototype detector available at CERN whose objective was to test and validate the technologies for the first DUNE-FD module based on the Horizontal Drift LArTPC technology. At the time of writing, the detector is being upgraded for the next data-taking phase. ProtoDUNE-SP is a 1 kton cryostat with an active volume of 7.2m x 6.0m x 6.9m (figure 3.6). The detector consists of 6 APAs separated by a central CPA wall. Figure 3.7 shows a schematic representation of the top down view of the ProtoDUNE-SP detector. The detector is split into three sections: upstream, midstream and downstream. The upstream section comprises the two APAs that are close to the beamline. In addition, the x, y, z coordinate system has the origin placed at the bottom corner of the upstream section, on the edge of the central CPA wall. The x coordinate increases with the beam's left side and the z coordinate increases towards the downstream section. The y coordinate increases vertically upwards. Finally, note that the beamline is also slightly angled (approximately 13°)

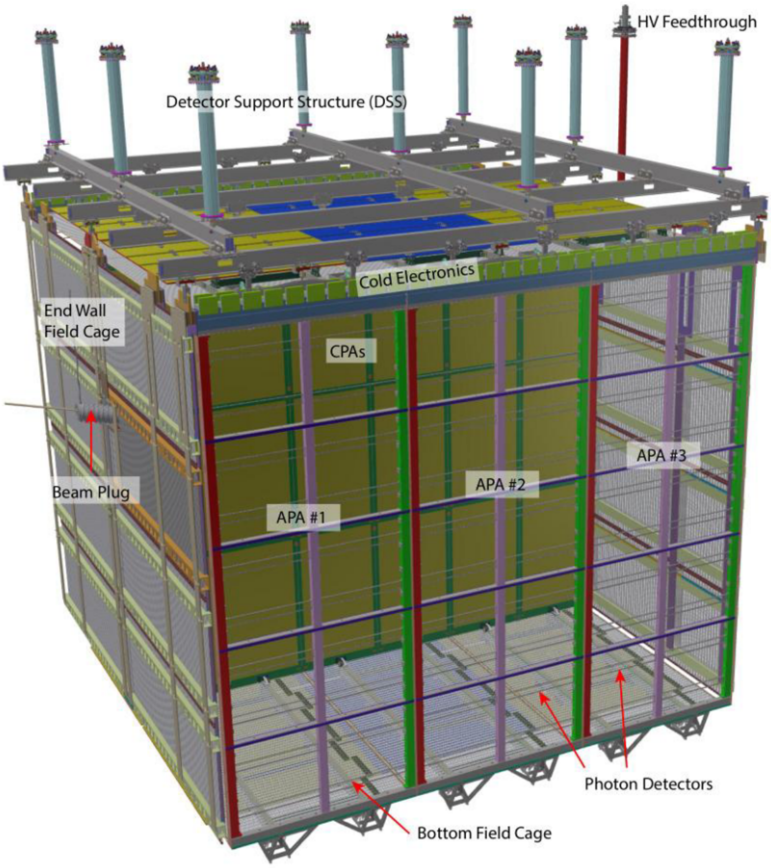


Figure 3.6: Diagram of the TPC components of the ProtoDUNE-SP detector. Figure taken from [10].

towards the beam right side in both the x-z and y-z planes.

ProtoDUNE-SP started taking data for the first time in 2018. It provided successful insights into the design and evaluation of large-scale LArTPC detectors in view of the much larger DUNE-FD detector. In particular, ProtoDUNE-SP represents a successful prototype for the DUNE-FD because it was designed to have similar drift lengths and operation conditions expected in the DUNE-FD Horizontal Drift module.

Note that other promising detector technologies, alternative to the Horizontal Drift, are being tested in the ProtoDUNE area at CERN. Examples of these LArTPC consist in detectors where a vertical drift of the ionization charge occurs compared to the typical horizontal drift across the active volume of liquid argon. Extensive research and development is being carried out at the time of writing to assess the suitability and the operation of such technologies for the other DUNE-FD modules.

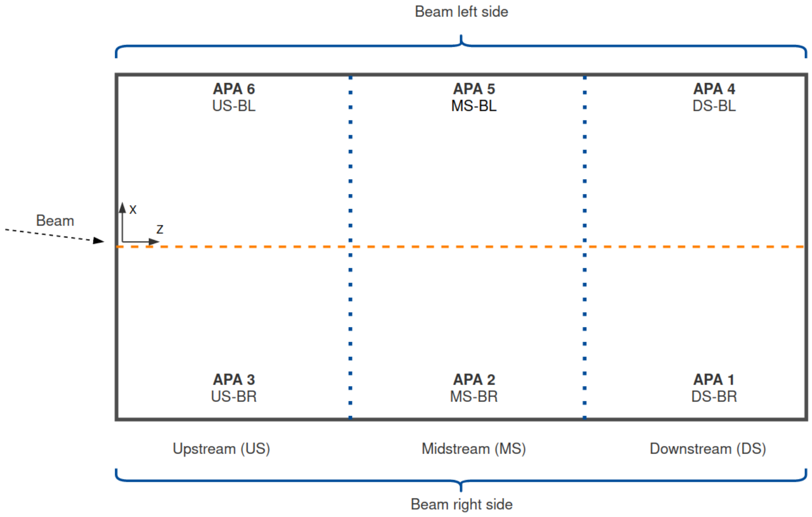


Figure 3.7: Schematic representation of the top down view of the ProtoDUNE-SP detector.

Cold Electronics

The drift field chosen for the ProtoDUNE-SP detector is 500 V/cm and each detector section provides two drift volumes with a distance of approximately 3.6 m. Each APA has a total of 2560 wires resulting in 15360 channels to be read out. Front End Mother Boards (FEMBs) are used to read out the signals from the wires of the APA. Each APA has 20 FEMBs located directly on top of the detector in order to be close to the wires and reduce the noise recorded by the electronics. The Cold Electronics (CE) system is responsible for collecting the signals from the APA wires and, then, amplifying, shaping and digitizing them before transmitting the processed signals to Warm Interface Boards (WIBs). These devices are interface electronics that handle the transmission of the signals to the data acquisition system. Further details on the data acquisition system of ProtoDUNE-SP will be illustrated in chapter 5.

Photon Detectors

Each APA frame is coupled with photon detectors used to collect the scintillation light produced by charged particles in the LAr. The scintillation light (photons with approximately 128 nm in wavelength) is converted into visible light by using wavelength shifters. Subsequently, the visible light is collected and converted into an electrical signal by an array of silicon photomultipliers. Note that in ProtoDUNE-SP different photon detector technologies have been tested to evaluate the best options to use in the DUNE Far Detector modules. The photon detectors are crucial elements because they allow the experiment to measure the start drift time t_0 (equation 3.2) and, therefore, they aid in the reconstruction of the neutrino interaction.

Cosmic Ray Tagger

A cosmic ray tagger (CRT) is a device used to provide triggers from cosmic rays. It is placed both upstream and downstream (along the z coordinate or beamline direction) of the ProtoDUNE-SP detector. In practice, the CRT is a scintillation counter that measures the x and y position of cosmic muons. By taking into consideration the coincidence hits registered by the CRT upstream and downstream it is possible to form tracks that are also matched with the reconstructed tracks from the TPC. In this way, the CRT can provide triggering capabilities and provide means to calibrate the detector.

4

Particle identification in LAr detectors

This chapter deals with the work carried out in the event reconstruction field and, in particular in the use of an innovative algorithm for track/shower hit classification based on Deep Neural Networks. Improving the track and shower hit classification is an important task in the event reconstruction chain that may have a crucial impact on the analyses. The ProtoDUNE-SP experiment was used as a testing platform to exercise the reconstruction algorithm in view of the future DUNE experiment.

4.1 Neutrino interactions in LAr

At the energies expected for the DUNE beam (few GeV), the dominant final state topology of a neutrino interaction with the argon will consist of a lepton, some mesons (e.g. pions, kaons) and possibly a number of baryons. These output particles have to be properly identified and their energies need to be measured in order to perform an accurate oscillation analysis.

As introduced in chapter 2, in an oscillation analysis dedicated to CP-violation it is important to distinguish the incoming neutrino flavor. Therefore, there is the need to separate electrons and muons from the hadronic backgrounds originating from all possible events. The typical example is the decay of neutral mesons (e.g. π^0) produced from NC interactions: they produce EM showers that can mimic the signature expected from ν_e-CC interactions. This kind of misreconstruction introduces a systematic error in the measured rate of the neutrino flavors in the Far Detector.

The identification and reconstruction of EM showers becomes then crucial for the analysis. From an experimental perspective, this translates into being able to correctly identify hits in the detector and determine if they originate from a particle producing a track or a shower. Having a good discrimination at the hit level is also important for the shower energy reconstruction which is largely based on measuring the total number of energy deposits, i.e. counting number of hits associated to a shower. Therefore, having an algorithm that is optimizing the efficiency of the shower hit identification can improve the energy resolution of the shower and, thus, of the event.

4.2 Neutrino event reconstruction and track/shower hit classification

Neutrino event reconstruction represents a critical task in oscillation physics analysis. In liquid argon TPC detectors (LArTPC), event reconstruction is usually accomplished by combining multiple algorithms to achieve the most accurate result.

Pandora is a general-purpose framework introduced for pattern recognition and is now used mainly in LArTPC experiments [12]. Pandora's approach consists of breaking the event reconstruction chain into smaller and well-defined tasks that specifically designed algorithms can handle. The complexity of such algorithms varies from simple energy cuts to more advanced machine learning tools. Examples of the tasks performed by Pandora consist in hit classification, hit clustering, vertex identification, track identification. For the DUNE Horizontal Drift detector, the hit is defined as the energy deposition collected by a single wire.

One critical component of the event reconstruction chain is the separation between track and shower hits. In ProtoDUNE-SP, the classification task of different particle types is done with Convolutional Neural Networks (CNNs) [13] where sections of the detector are converted into 2D images and then classified. However, one of the challenges of LArTPC detectors is that they produce a large amount of sparse and locally dense data, which becomes challenging to handle for larger detector volumes. The following sections will discuss the usage of an innovative algorithm specifically designed to handle sparse data on three-dimensional space points.

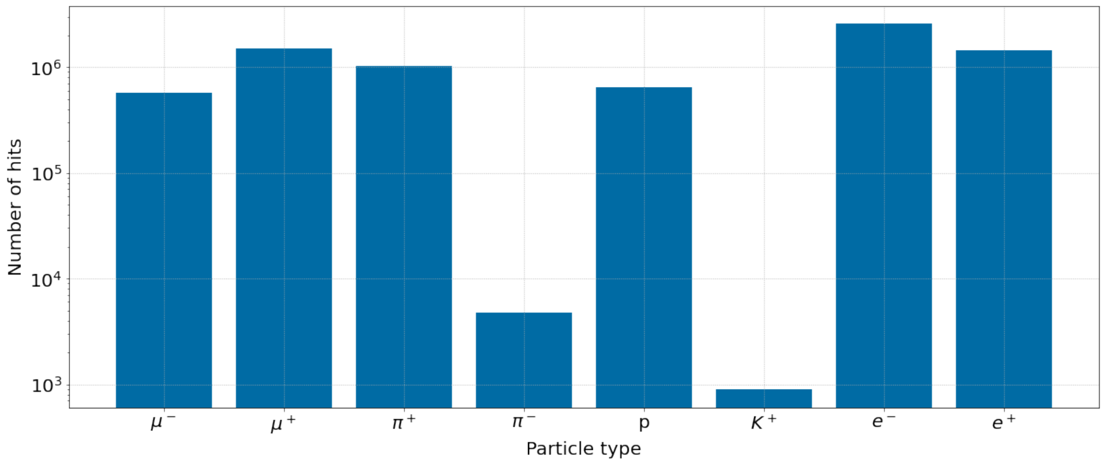


Figure 4.1: Distribution of particle types in the simulated MC dataset that is used for the feature evaluation. Most of the shower hits originate from electrons and positrons. Tracks are associated to muons, pions and protons. Number of hits on the y-axis expressed in a logarithmic scale.

To distinguish and classify track and shower hits a set of features were extracted from simulated Monte-Carlo (MC) ProtoDUNE-SP data. This was done taking advantage of the 3D nature of the problem to achieve the classification between hits originating from a track or a shower particle. The ultimate goal is to utilize as many relevant features as needed to train a Deep Learning model specifically designed for the classification task.

4.2.1 Dataset evaluation

The MC data samples used for the feature extraction consist of ProtoDUNE-SP datasets. More than 7M hits have been processed with events containing a mixture of track-producing particles (muon, pion, proton and kaon) and EM shower-producing particles (positron and electron). Note that a modified version of the code of the DUNE Convolutional Visual Network [13] was used to process the necessary features from the simulated data.

Figure 4.1 shows how the number of hits for the various particle types are distributed in the input dataset (number of hits expressed in a logarithmic scale). As it can be noted, most of the shower hits originated from positron and electron particles whereas track hits are mostly associated with muons, pions, and protons. There is also a small contribution of positive kaons (K^+) and negative pions (π^-) in the dataset.

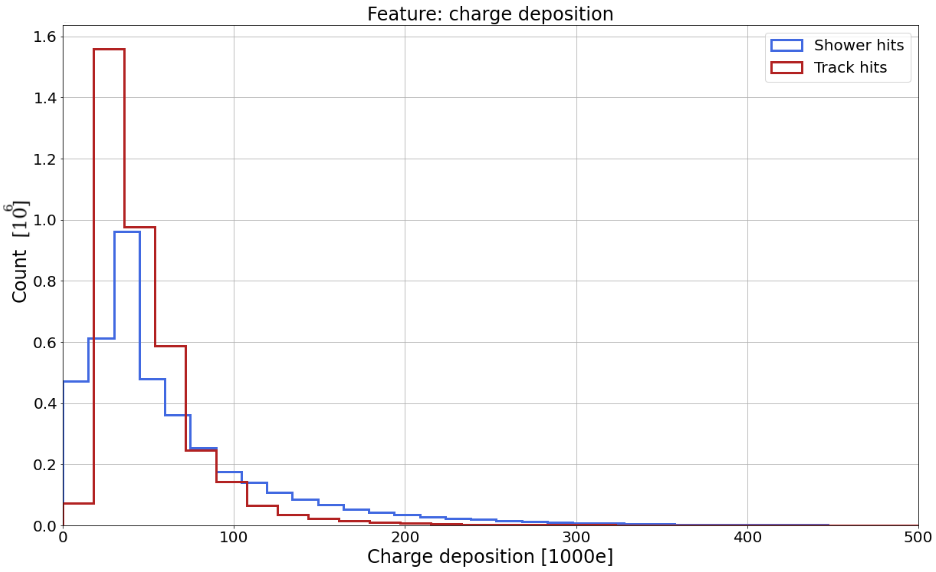


Figure 4.2: Charge distribution of the track and shower hits in the simulated MC dataset.

4.3 Feature extraction

The first feature that was considered is the charge deposited by individual hits. The rationale for using the charge is that track hits (pions and protons) are expected to lose more energy as they traverse the detector volume compared to the shower hits (electrons, positrons). Figure 4.2 shows the charge distribution for both track and shower hits in the simulated MC dataset.

The following two features that have been selected refer to the properties of hits based on their proximity to other neighboring hits. Specifically, the angle and the dot product were computed between the relative positions formed by connecting the selected hit with the two closest hits.

Figure 4.3 shows the angle distribution between a hit and the two closest ones for both track (red) and shower (blue) hits. The angle is a useful discriminating feature between track and shower hits. This is because shower hits are more isotropically distributed compared to track hits that are closely aligned with each other. This behaviour is reflected in figure 4.3 where track hits are peaked at angles of 0 and π, suggesting that neighboring hits are aligned with each other.

Figure 4.4 shows the distribution of the dot product between a given hit and the two closest ones. The figure suggests that the distribution of track hits (red) is wider than the shower hits (blue) distribution. Therefore, the dot product distribution can also be another useful feature

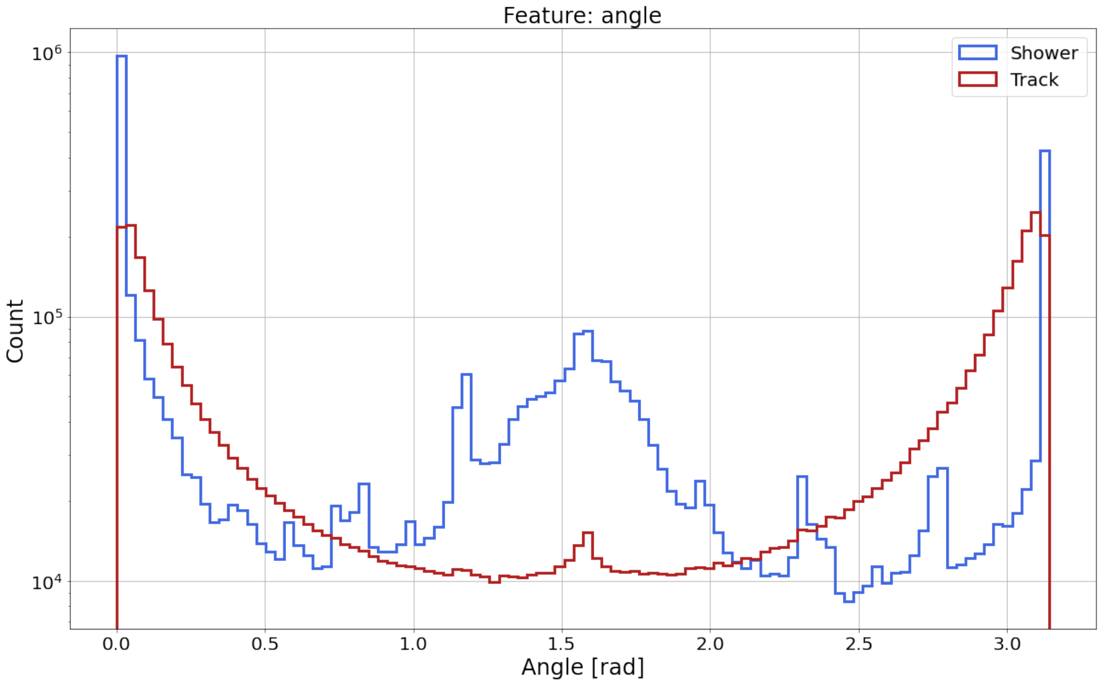


Figure 4.3: Distribution of the angle between a hit and the two closest ones for track and shower hits from the simulated MC dataset.

used in the classification task. Finally, note that the angle (θ) and the dot product between two vectors **a** and **b** are two properties not necessarily correlated with each other as it may seem at first glance given that $\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\|\|\mathbf{b}\|cos(\theta)$. This is because the magnitude of the two vectors **a** and **b** are not related to the angle θ between the two vectors. Therefore, both the angle and the dot product can be useful for distinguishing between track and shower hits. Note that events have been pre-selected with at least 3 hits to be able to evaluate both the angle and the dot product features.

The next set of features that was used in the classification task is the number of hits within a certain configurable distance. The rationale for using this discriminating feature is that shower hits tend to have a wider distribution than track hits. Three distance ranges (3 cm, 10 cm, 30 cm) have been selected as they yield the best discrimination for track and shower hits (further details in section 4.4). Figures 4.5 shows the ratio of the distributions of the number of neighboring hits at a distance of 3 cm and 30 cm. The two figures highlight that shower hits have a wider distribution than track hits. These, in particular, are centered at a value of 0.1 which represents the ratio of the number of neighboring track hits between 3 cm and 30 cm.

Finally, the last set of features that was computed is the total charge deposited by the hits within a spherical distance R. The reason why the charge over distance represents an interesting feature

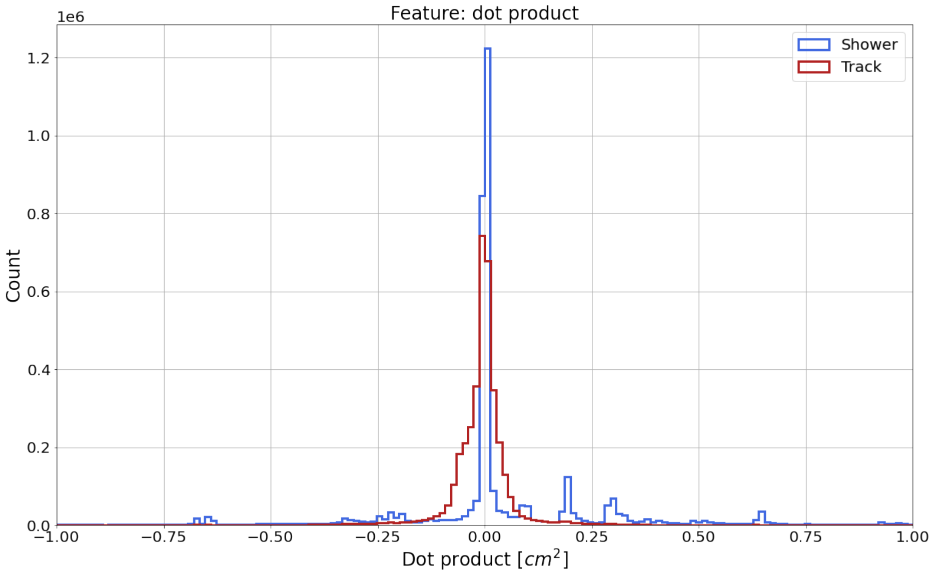


Figure 4.4: Distribution of the dot product between a hit and the two closest ones for track and shower hits from the simulated MC dataset.

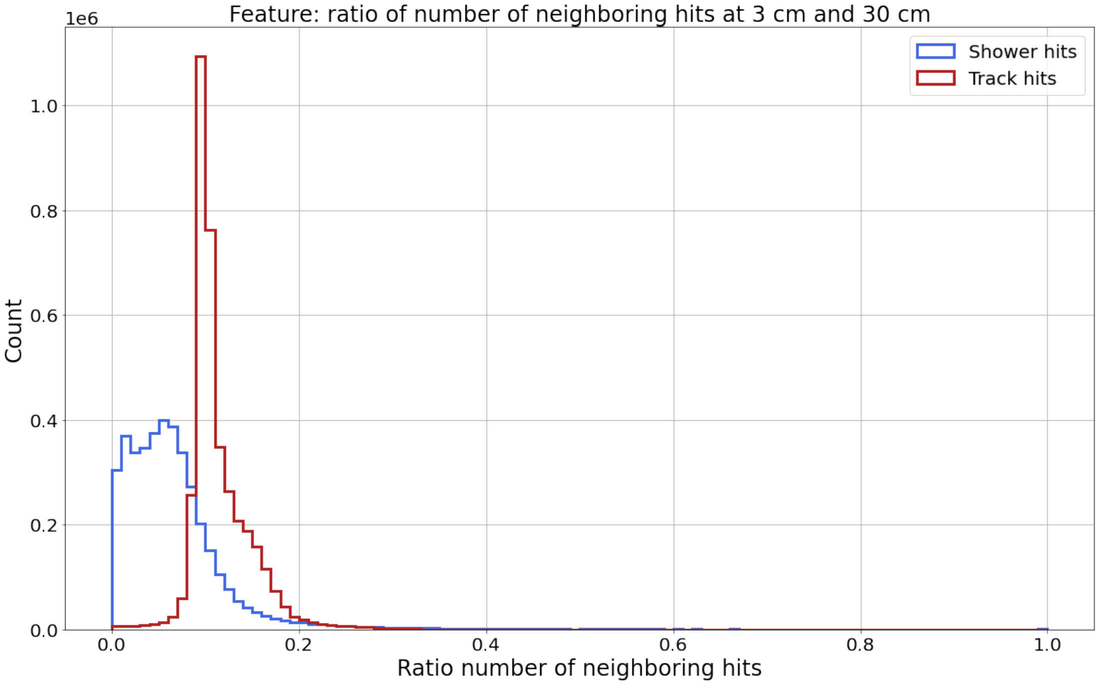


Figure 4.5: Ratio between the distribution of the number of neighboring hits at a distance of 3 cm and 30 cm for both track and shower hits.

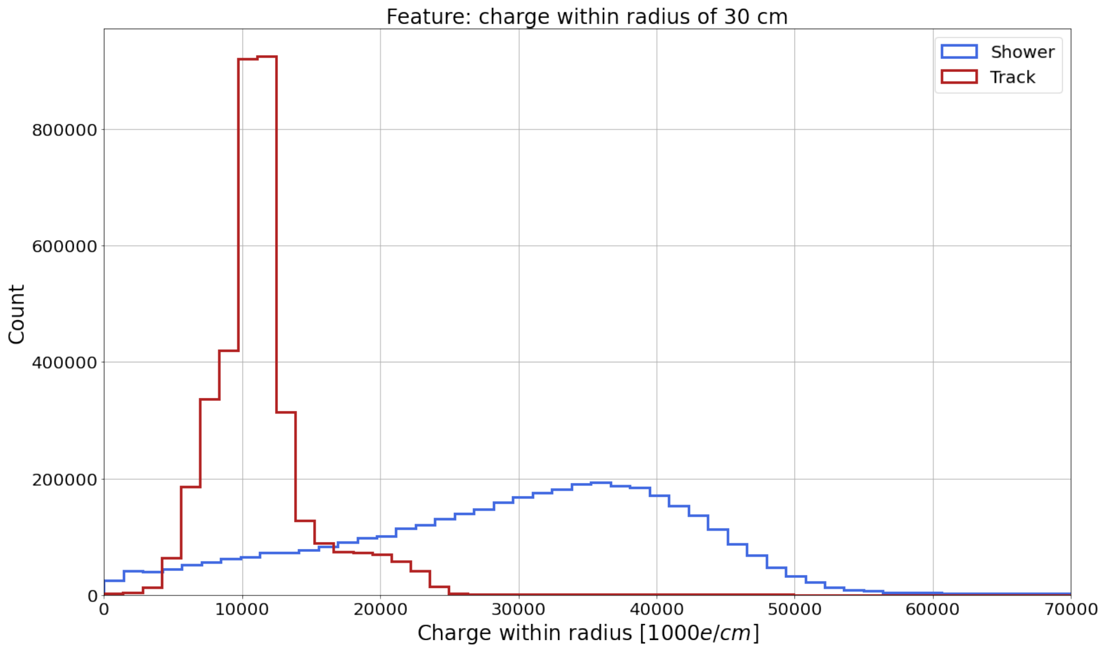


Figure 4.6: Distribution of the charge deposited by track and shower hits within a distance of 30 cm .

for discriminating track hits from shower hits is because it is a property very similar to the commonly used dE/dx for particle identification. In fact, the charge released by the incoming particles in the detector volume is proportional to the energy deposited. Figure 4.6 shows the distribution of the charge deposited by track and shower hits within a distance of 30 cm. There is a clear separation between the peaks for both track and shower hits which suggests that the charge over a certain distance is another useful parameter for the classification task.

In total, nine quantities were selected for the classification task: charge deposition of each hit; angle and dot product between two neighboring hits; the number of neighboring hits as well as the total charge within a distance of 3 cm, 10 cm and 30 cm. The distribution of these quantities offers discriminating power between track and shower hits and, therefore, they were used as input into a Deep Learning model. Note that processing the number of neighboring hits is a computing-intensive task. Therefore, for convenience, both the number of neighboring hits and the charge deposition within a certain distance were computed with the same set of lengths.

4.4 Submanifold Sparse Convolutional Neural Network

Submanifold sparse convolutional neural networks (SparseNet) [14] are a class of Deep Learning methods primarily designed for 3D image reconstruction, image completion or semantic segmentation problems. In [14], the SparseNet has proved to be quite effective when dealing with sparse data and potentially minimal resource utilization compared to other classes of Convolutional Neural Networks. The three-dimensional hits produced in ProtoDUNE-SP are well suited for the SparseNet algorithm as they are locally dense and sparsely located in the detector volume. Moreover, the computational benefit of utilizing the SparseNet may also be more relevant for larger LArTPC detectors such as the planned DUNE experiment.

4.4.1 Architecture

Submanifold Sparse Convolutional Neural Networks (SparseNet) are convolutional neural networks that use generalized convolution operations on sparse tensors [14]. In the case of the SparseNet used for this research, the input tensor is a 3x3 grid that is associated with a 9-dimensional feature vector containing the features for the classification task.

For this research, the SparseNet was used to perform a two-label classification task for distinguishing between track hits and shower hits. A softmax activation function and a Stochastic gradient descent (SGD) optmizer have been used in the SparseNet. The network's output is a score ranging from 0 for track objects and 1 for shower objects. In addition, the Minkowski Engine [15] was used to support the operations of convolution and pooling that are needed by the computation with sparse networks.

4.4.2 Data samples - MC and data

The ML training and validation was performed on a MC dataset of approximately 2.5M hit entries. Each entry contains the 3-dimensional coordinate position of the hits (x,y,z), the hit truth value (i.e. the PDG value of the particle originating the hit) and the selected nine features as discussed in the previous sections. Note that the PDG value of the particle, in principle, can also be used to train the network for different tasks such as performing particle identification (e.g. classify only hits originating from photons and electrons). For the classification task, the PDG values of muons, pions, protons and kaons have been assigned to a track output because they typically produce a track in the detector. Similarly, electrons, positrons and photons have been assigned to a shower output. Finally, a smaller and independent data sample of 500k hits was also produced for the inference process.

The ProtoDUNE-SP dataset used for the evaluation consists of two runs from ProtoDUNE-SP taken with a 1 GeV/c beam momentum: run 5387 and run 5809. The selected runs contain both cosmic rays and charged particles from the incoming beam. In particular, run 5387 was taken with a trigger where positrons were vetoed. As a consequence, the dataset contains primarily π^+, μ^+ and protons. On the other hand, run 5809 was taken with a beam trigger which in turn results in a dataset consisting of mainly beam positrons. A total of more than 300k hits were processed with the same nine features used for the MC datasets. Overall, the selected ProtoDUNE-SP runs provide a balanced dataset containing both shower particles from the beam positrons and track particles mainly from pions and cosmic ray muons.

4.4.3 Evaluation of the SparseNet classification model

The first step of the ML evaluation consists in splitting the MC dataset. Three sets were prepared: training (60% of the total), validation (20% of the total) and testing (20% of the total). The training dataset was used for fitting the model, the validation dataset was used to provide an unbiased evaluation of the model while training and finally the testing dataset was used to assess the performance of the model after the training process. The objective was to use most of the dataset for training task and the remaining sample of hits for validation and testing. In addition, the training process of the SparseNet was done for a total of ten epochs (one complete pass through the training data) to avoid overfitting the model.

One of the scores commonly used in classification problems is the accuracy which represents the ratio of correct predictions with respect to the total number of predictions. For a binary classification problem (positive vs negative classes), the accuracy is given as:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP, TN, FP and FN represent respectively the number of True Positives (number of elements correctly labeled as belonging to the positive class), True Negatives (number of elements correctly labeled as belonging to the negative class), False Positives (number of elements incorrectly marked as belonging to the positive class) and False Negatives (number of elements incorrectly labeled as belonging to the negative class). In the case of the SparseNet used for this evaluation, the positive class refers to the track particles and the negative class refers to the shower particles.

Figure 4.7 shows the accuracy of the model as a function of the number of epochs for both the training and validation steps. After the training processes, the overall accuracy reached is more than 95% which demonstrates good discrimination power between track and shower

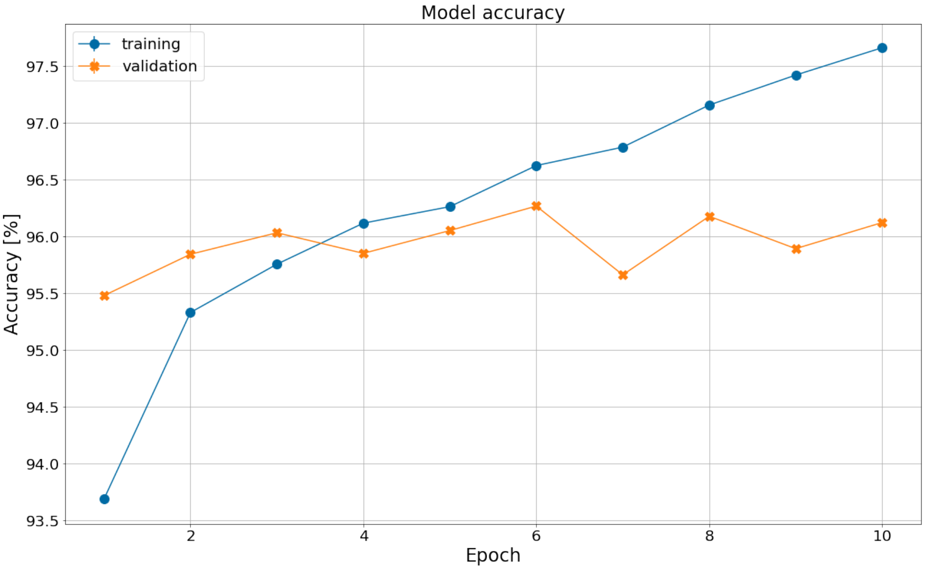


Figure 4.7: Accuracy as a function of the number of epochs for the SparseNet model for both the training and validation steps. Statistical uncertainty on the accuracy is in the orders of $0.01\ \%$.

hits. The statistical uncertainty on the accuracy is calculated as following assuming a binomial distribution for the hits belonging to the true or negative classes:

$$\sigma^2_{\mathrm{A}} = \frac{A(1-A)}{N} \tag{4.1}$$

where A is the accuracy and N is the total number of hits used in the evaluation. Note that the scale of the plot in figure 4.7 does not allow to appreciate the statistical uncertainty on the accuracy which is in the orders of 0.01%. The accuracy obtained for the SparseNet shows that the algorithm is capable of outperforming the currently deployed CNN in ProtoDUNE [16]. In fact, the results for the CNN show that on MC data the accuracy obtained is 87.3% [16].

The accuracy result is not sufficient for a good classification model because it does not indicate how small are the number of incorrectly identified classes. Therefore, more performance metrics are needed to fully evaluate the classification algorithm.

Purity and efficiency are the other two performance metrics commonly used when evaluating a classification model. The purity for the positive class represents the fraction between the number of true positives with the total number of instances labeled as belonging to the positive class. On the other hand, the efficiency for the positive class is defined as the number of true positives

Table 4.1: Table summarizing both the purity and the efficiency of the SparseNet on the MC inference dataset.

Class	**Purity [%]**	**Efficiency [%]**
Track	96.8	97.8
Shower	93.5	90.7

divided by the total number of instances that belong to the positive class. By combining the information given by both the purity and the efficiency, it is possible to understand more in detail if a classification algorithm can correctly identify the right classes and provide a lower number of misidentified instances.

$$\text{purity (positive class)} = \frac{TP}{TP+FP}$$

$$\text{efficiency (positive class)} = \frac{TP}{TP+FN}$$

Table 4.1 shows the purity and the efficiency of the SparseNet for the track and shower classes when applied to the testing dataset. The results show that the purity of the two classes is above 90%, indicating that the percentage of misidentified hits belonging to a track (or shower) object is low. In addition, having an efficiency of over 90% also suggests that the algorithm can select most of the relevant hits belonging to a track (or shower) class. Uncertainty on these results can be safely ignored for the classification metrics as they are computed on a large dataset comprising millions of entries.

4.4.4 Tuning of the classification threshold

As mentioned above, the output of the SparseNet is a value between 0 and 1. A selection threshold is a value, between 0 and 1, used to separate the two classes under investigation. When developing a classification algorithm, it is important to select the threshold that maximizes the selection of the true classes and that minimizes the number of misidentified instances. This can done by varying the classification threshold value, retraining the model, performing the inference and counting the number of true and misidentified elements in the dataset.

One way to study the classification threshold is shown in figure 4.8 which represents the multiplication of the purity and efficiency for both track and shower classes. This is obtained by modifying the classification threshold and counting the number of true and misidentified elements and computing the purity and efficiency for each class and for each threshold. The figure

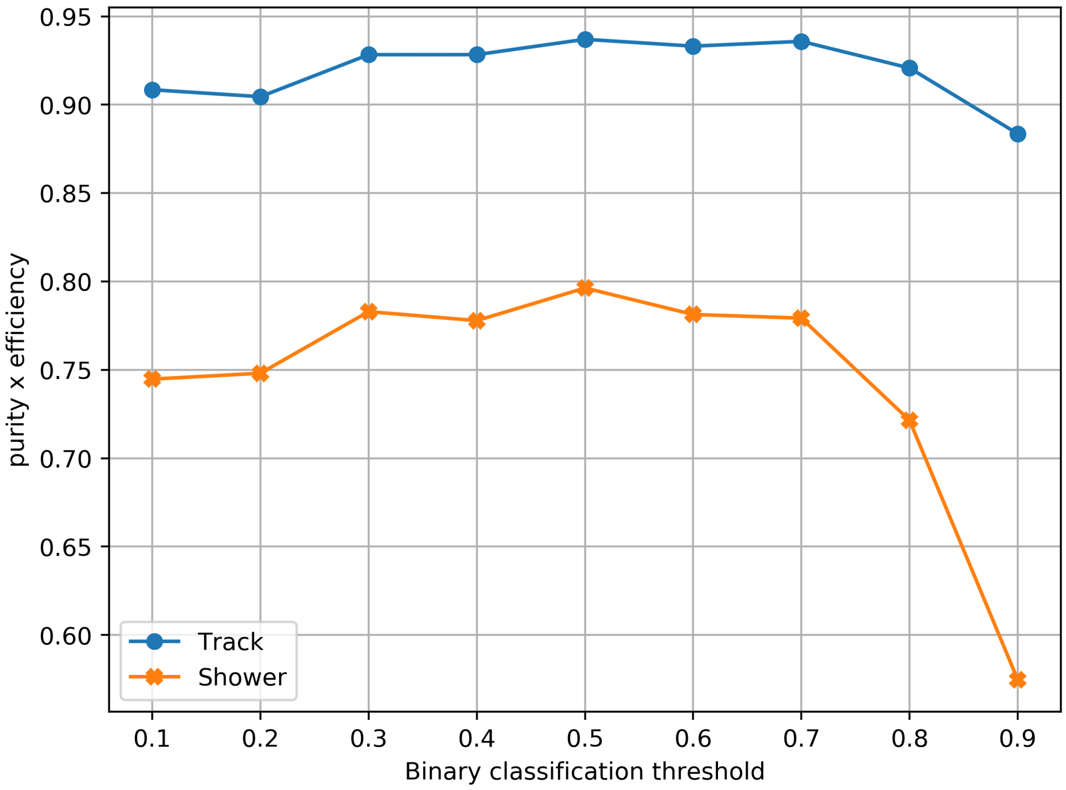


Figure 4.8: Multiplication of the purity and efficiency for both track and shower classes as a function of the classification threshold.

illustrates that a threshold cut of 0.5 represents the value that maximizes the selection of hits that belong to either a track or a shower while at the same time keeping the number of misidentified hits low. Note that the selection threshold used for the accuracy, purity and efficiency metrics discussed in the previous section was set to 0.5.

4.4.5 Particle-based performance metrics for the network evaluation

The ML performance metrics described in the previous sections can only be applied to the hits on which the model is used and do not consider the particle that produces the hits. Therefore, an average score was also computed by grouping all the hits belonging to a single track (or a single shower) object. In this way, it is possible to compute how well the network can correctly classify tracks (or showers) rather than relying solely on the hit classification. These quantities are called track and shower scores and they are computed as following:

1. Identify all the hits belonging to a single track (or shower)

2. Count the number of hits correctly identified by the network

3. Compute the efficiency by taking the ratio between the number of correctly identified hits and the total number of expected hits for a given track (or shower)

4. Repeat the steps above for all the particle objects in the dataset

5. The score is given by computing the arithmetic mean of the efficiencies obtained for all the track (or shower) particles in the dataset

The MC truth information for each hit was used to ensure that the given hits belonged to a specific track or shower. Specifically, a track identifier was used to match the hits originating from the same track and a particle identifier was used to match the hits originating from a shower-producing particle.

The inference dataset was used for the track and shower score evaluation. This is a dataset produced in the same way as the training and validation samples and it was used to provide an unbiased evaluation of the model after it has been trained.

The average track and shower scores obtained for the inference dataset:

$$\text{Track score: } 0.852 \pm 0.008$$

$$\text{Shower score: } 0.829 \pm 0.008$$

Note that the uncertainty on the scores is given by the variance of the mean: $\sigma_\mu = \frac{\sigma}{\sqrt{N}}$ where N is the total number of tracks (or showers) and σ is the standard deviation of the efficiencies computed for each track (or shower). Specifically, the standard deviation of the efficiencies for the track hits is approximately 0.245 and for shower hits is approximately 0.157. Figure 4.9 shows the distribution of the efficiencies calculated for the track score. As it can be noted, most of the tracks have a track efficiency closer to 1, showcasing the good performance of the SparseNet in correctly identifying most of the hits that belong to a single track.

The newly introduced average scores provide a useful indicator of how the network is performing in the classification of the entire track and shower objects: if the score is closer to 1 then the network has correctly identified most of the hits belonging to a track or a shower particle. It is also interesting to investigate how the track and shower score change as a function of the minimum number of hits that constitute a track or a shower particle. This is due to the fact that a particle that has deposited, for example, only three hits in the detector does not represent

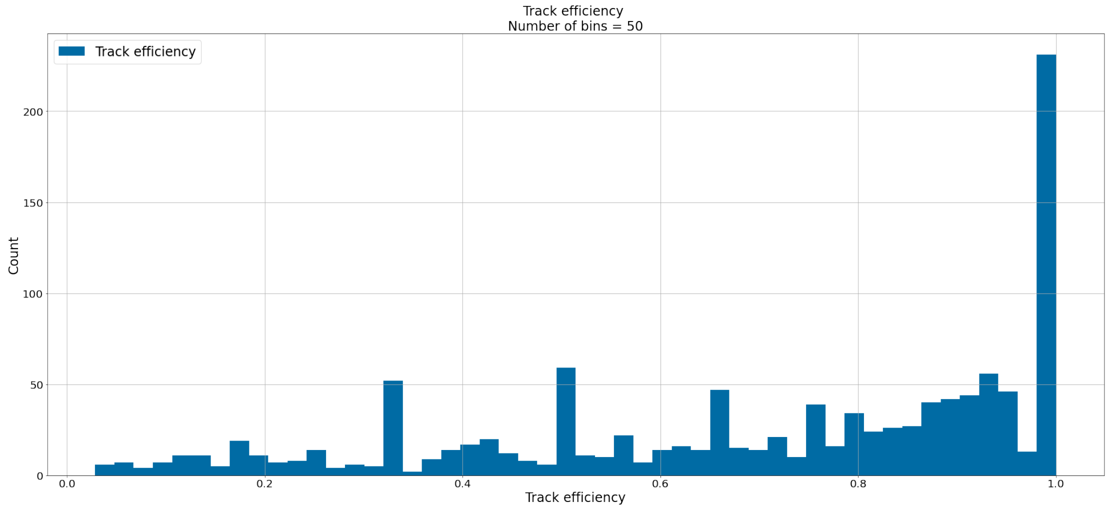


Figure 4.9: Distribution of the track score efficiency.

an interesting track or shower but may lead to a lower score if the network did not correctly identify all the hits. Figure 4.10 shows how the track score is affected by the minimum number of hits per track. As it can be observed, the track score increases from approximately 85% to more than 92% by requiring at least ten hits for each track. In addition, this also means that there are many short tracks (i.e. tracks with a small number of hits) where not all of the hits have been correctly identified by the network. Finally, for completeness, figure 4.11 shows the distribution of the shower score efficiency for all the shower events in the dataset. Contrary to the behavior for the track score efficiency (figure 4.10), in this case, the shower score tends to decrease when requiring a minimum number of hits per shower. This is because the network is more likely to correctly predict all the hits belonging to a shower with a small number of hits compared to a shower with, for example, ten hits.

The track and shower scores represent interesting particle-based metrics that are very useful when evaluating the track/shower hit classification algorithm. In fact, they give an insight into the network's overall performance when applied to all the hits of a particle producing a track or shower event. By considering the track and shower scores, the evaluation of the network training as a function of different input features was also conducted. Table 4.2 shows the summary of the track and shower scores for different training scenarios for the SparseNet model. Not surprisingly, including all the features in the training process yields the best scores when using the testing dataset. Excluding just the charge over distance features or both the charge over distance and the number of neighboring hits over distance leads to a reduction of the track score up to 7% and the shower score up to 8%.

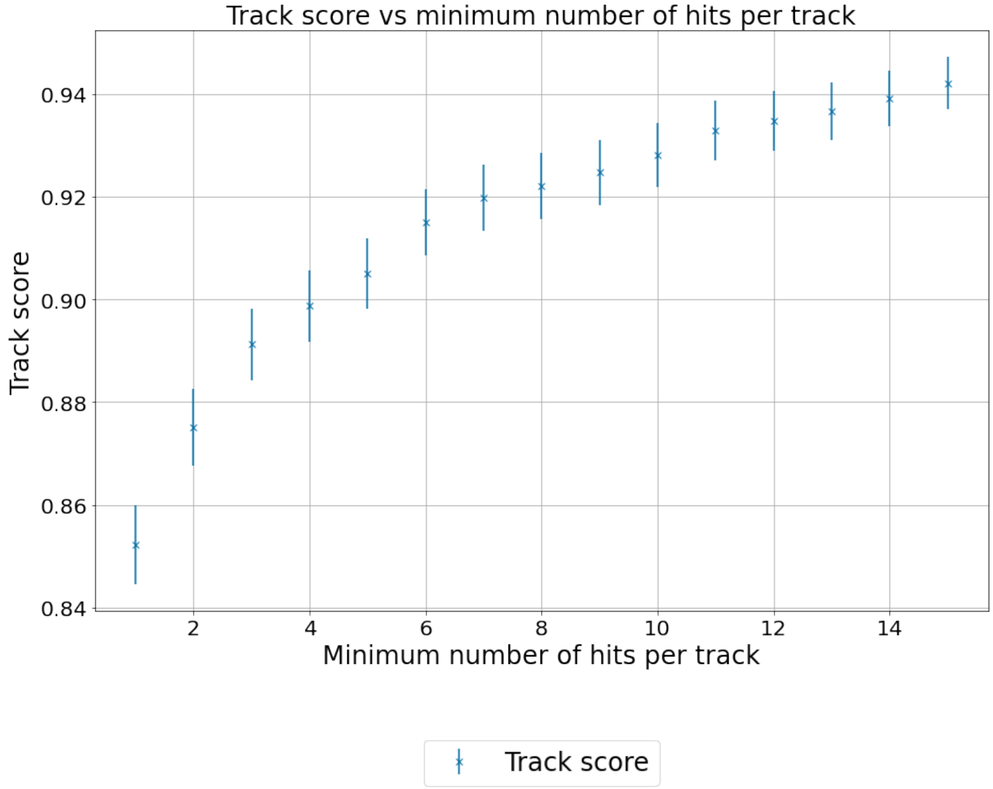


Figure 4.10: Track score as a function of the minimum number of hits that make a track.

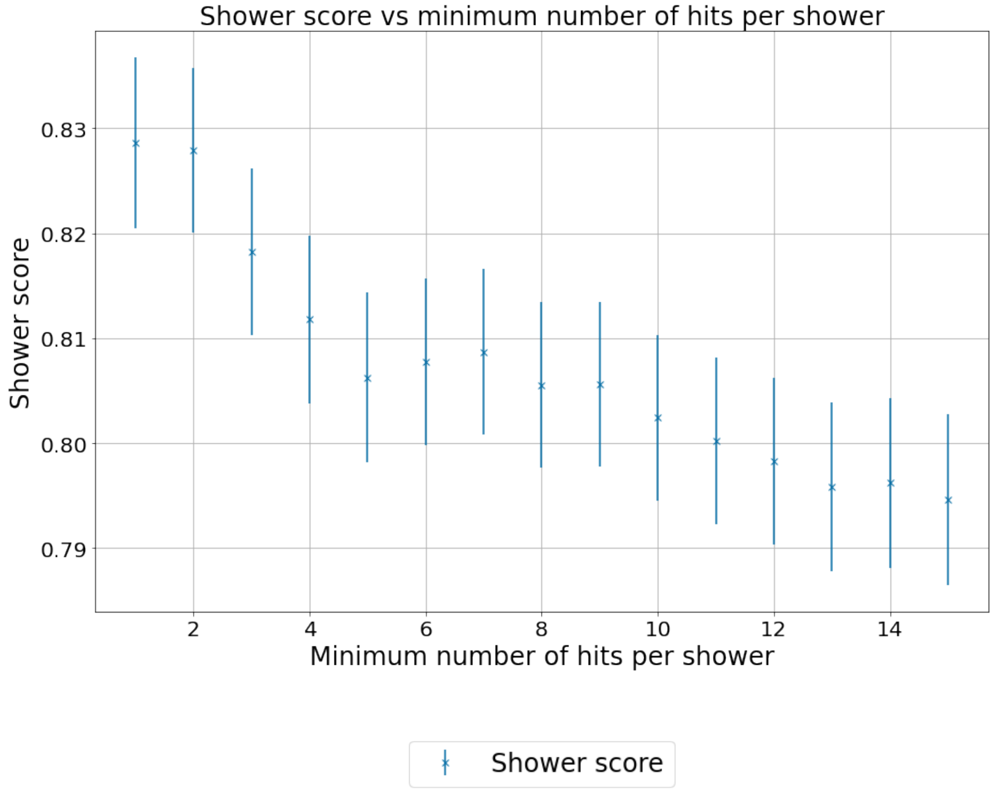


Figure 4.11: Distribution of the shower score efficiency for all the shower events in the dataset.

Table 4.2: Summary of the track and shower scores for different training scenarios for the SparseNet model. Using all the nine features for the training yields the best results in terms of both track and shower scores.

Model type	**Track score**	**Shower score**
9 features (all included)	0.875 ± 0.003	0.813 ± 0.008
6 features (exclude charge over distance)	0.866 ± 0.003	0.804 ± 0.007
3 features (exclude both charge over distance and number of neighboring hits over distance)	0.807 ± 0.002	0.732 ± 0.009

4.5 Evaluation of the SparseNet in ProtoDUNE-SP MC events

Beyond testing performance metrics, it is also equally important to evaluate the response (or the score) of the network in a machine learning model. This is done by applying the network on a dataset where the truth of both classes (track and shower hits) is known. In this way, it is possible to evaluate if some hits have been incorrectly identified as belonging to one or the other classes. As outlined in section 4.4.1 the SparseNet gives in output a value between 0 and 1 which represent respectively the track and shower classes. In reality, since the network was designed for a binary classification problem, the output for each hit are two scores, both between 0 and 1, that represent whether the hit belongs to a track or shower class. Figure 4.12 shows the distribution of the SparseNet score for the shower class when the network is applied to the inference dataset. Two distributions are shown: track (red) and shower (blue) hits. Two peaks are observed, one at around 0 for the track hits and one at around 1 for the shower hits. This reflects the fact that the network has correctly classified the majority of the track and shower hits and it has associated them with the correct class labels. Interestingly, for the distribution of the track hits there is a smaller peak at around 1, more than two orders of magnitude lower than the one at zero, which reflects that some of the track hits have been incorrectly labeled as shower hits.

An analysis of the network's output was also performed on track and shower hits originating from different particles. Figure 4.13 shows the distribution of the SparseNet score for the shower class for both track and shower hits originating from pions, muons, positrons and electrons. As an example, the distribution of the SparseNet score for the shower class for positron (violet curve) and electron (light blue) hits is peaked at a score of 1, which is the network representation of a shower. On the other hand, the distribution of the SparseNet score for hits originating from pions and muons have the opposite behavior. In fact, pions and muons produce a track-like energy deposit inside the detector and, therefore, they are observed with a SparseNet score

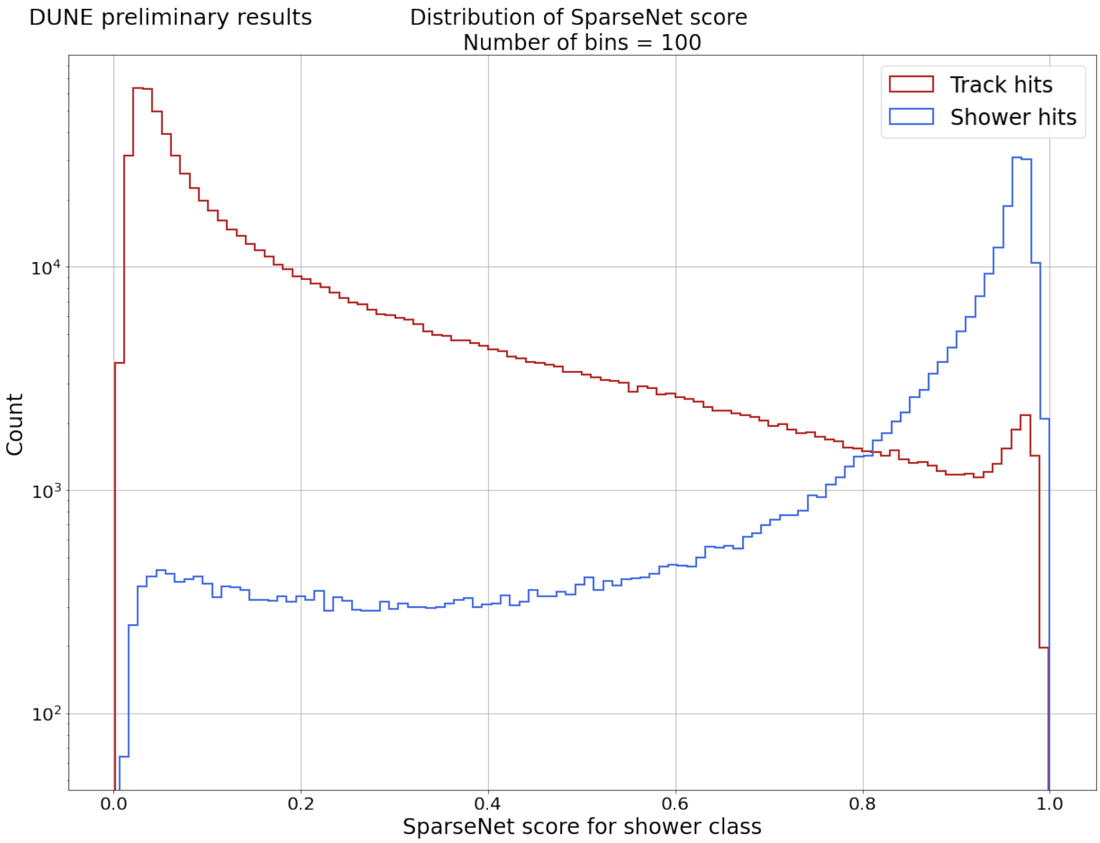


Figure 4.12: Distribution of the SparseNet score for the shower class for both track and shower hits.

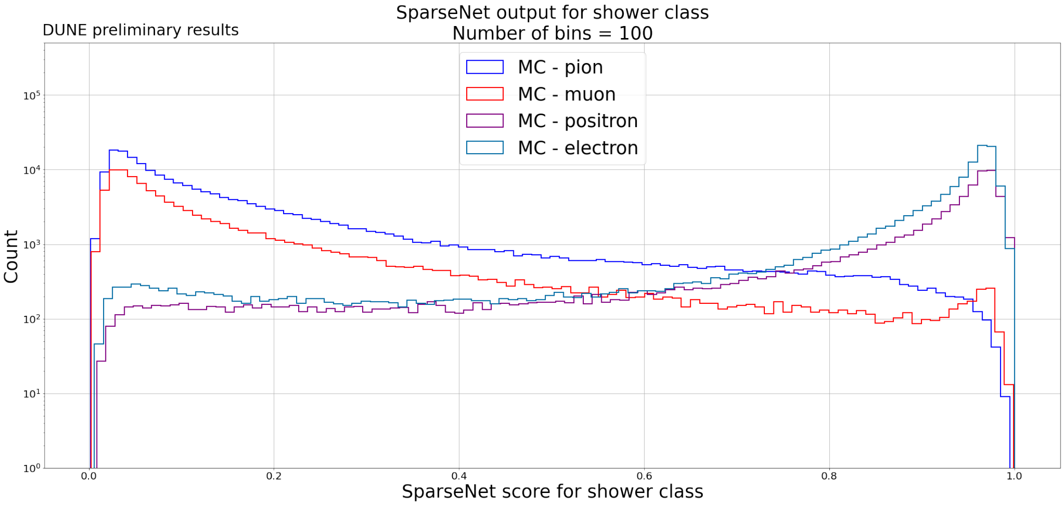


Figure 4.13: Distribution of the SparseNet score for the shower class for both track and shower hits from different particles. Pion, muon, positron and electron hits are shown.

Table 4.3: Table summarizing both the pseudo-purity and the pseudo-efficiency of the SparseNet on the ProtoDUNE-SP dataset.

Class	Pseudo-purity [%]	Pseudo-efficiency [%]
Track	97.7	97.8
Shower	95.6	95.3

distribution (figure 4.13) peaked at zero.

4.6 Evaluation of the SparseNet in ProtoDUNE-SP data events

Preliminary results show excellent performance of the SparseNet model when it is applied to data runs from the ProtoDUNE-SP detector (runs 5387 and 5809). In this case, the prediction of the network is compared with the results from the event reconstruction software (Pandora) since the truth information is not available for the ProtoDUNE-SP dataset as compared to the MC samples. Therefore, the presented performance metrics are not true purities and true efficiencies because they assume that Pandora is 100% efficient. To distinguish between the metrics used for the MC evaluation and the ones used for the ProtoDUNE-SP data evaluation, in this section the metrics will be referred as pseudo-purities and pseudo-efficiencies. This is a reasonable approach considering that Pandora's efficiency is greater than 95% for track and shower particles made of more than 100 hits [17]. The efficiency of Pandora reaches 99% for tracks made with more than 400 hits.

The psuedo-purities and pseudo-efficiencies obtained with the ProtoDUNE-SP data are illustrated in table 4.3. The results show that both the track and shower classes have values for the purity and efficiency greater than 95%.

Figure 4.14 shows the distribution of the SparseNet score for the shower class when the network is applied to the ProtoDUNE-SP dataset. Two distributions are shown: track (red) and shower (blue) hits. As it can be observed track hits have been correctly classified by the network and with very few misidentified hits that have been classified as showers. On the other hand, there is a peak at zero in some shower hits, which suggests that these hits have been incorrectly classified as track-like hits.

Overall the SparseNet algorithm provides a satisfactory separation between track and shower hits. A visual illustration of the use of the SparseNet on ProtoDUNE data is shown in figures 4.15 and 4.16 which represent a ProtoDUNE-SP dataset from Run 5809 before and after ap-

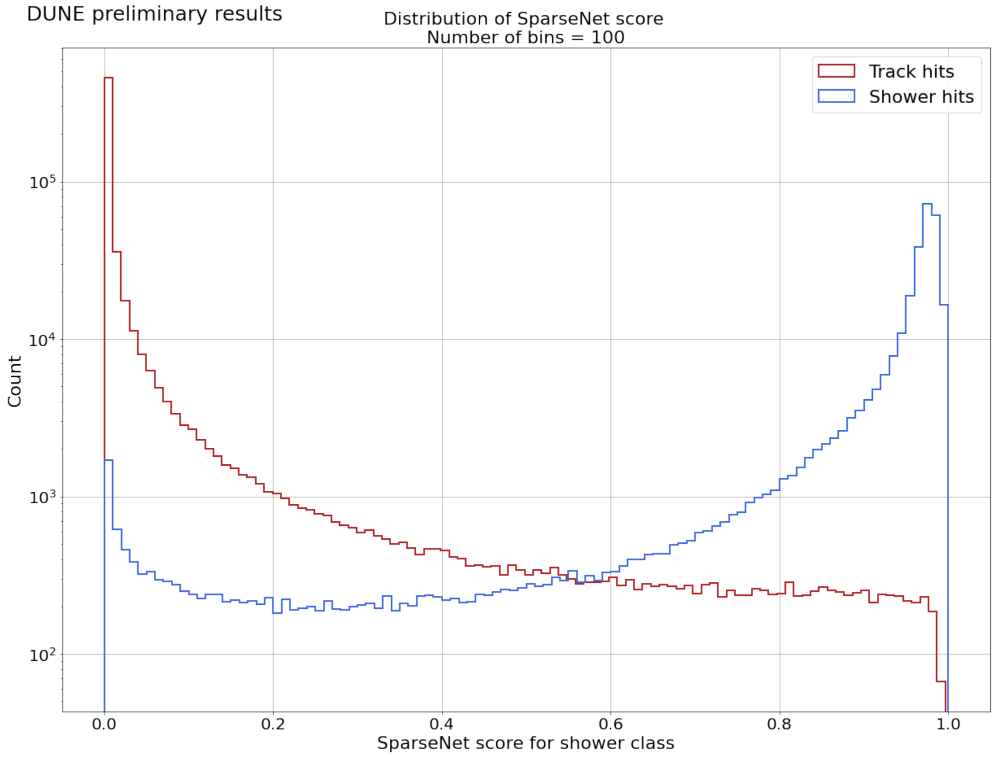


Figure 4.14: Distribution of the SparseNet score for the shower class for both track (red curve) and shower hits (blue curve) using the ProtoDUNE-SP dataset. Most of the track and shower hits have been correctly identified (peaks at 0 and 1). A small peak at zero on the blue curve suggests that a fraction of the shower hits have been incorrectly classified as track hits.

plying the SparseNet. The natural next step of the SparseNet is to integrate the classification algorithm into the ProtoDUNE-SP analyzer framework and test its performance on an end-to-end analysis. At the time of writing the SparseNet model has been integrated within the ProtoDUNE analysis framework.

4.7 Conclusion

This chapter has shown the development and testing of a classification algorithm between track and shower hits. This is a crucial step in the neutrino event reconstruction and for the oscillation analysis. Initially, a set of input parameters was carefully selected for the classification task by studying their distributions and comparing their behaviour on track and shower hits. The selected parameters were the hits charge; angle and dot product between two closest hits; number of hits and charge deposition across a range of three distances.

In particular, this work has shown the advantages of using a novel Deep Learning model that leverages convolutions on sparse data (SparseNet). The SparseNet has been designed for the

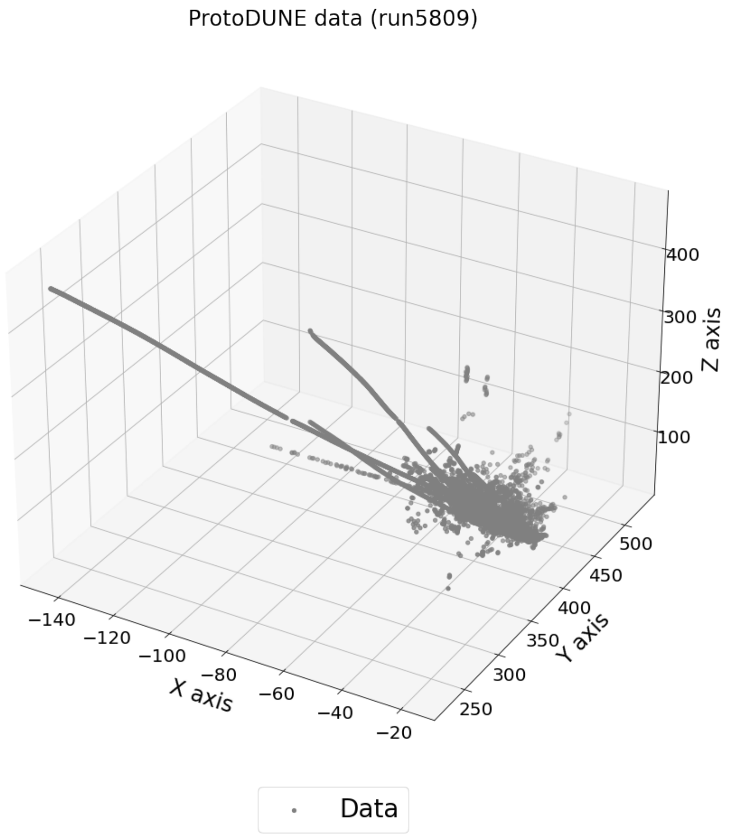


Figure 4.15: Dataset from ProtoDUNE-SP Run 5809.

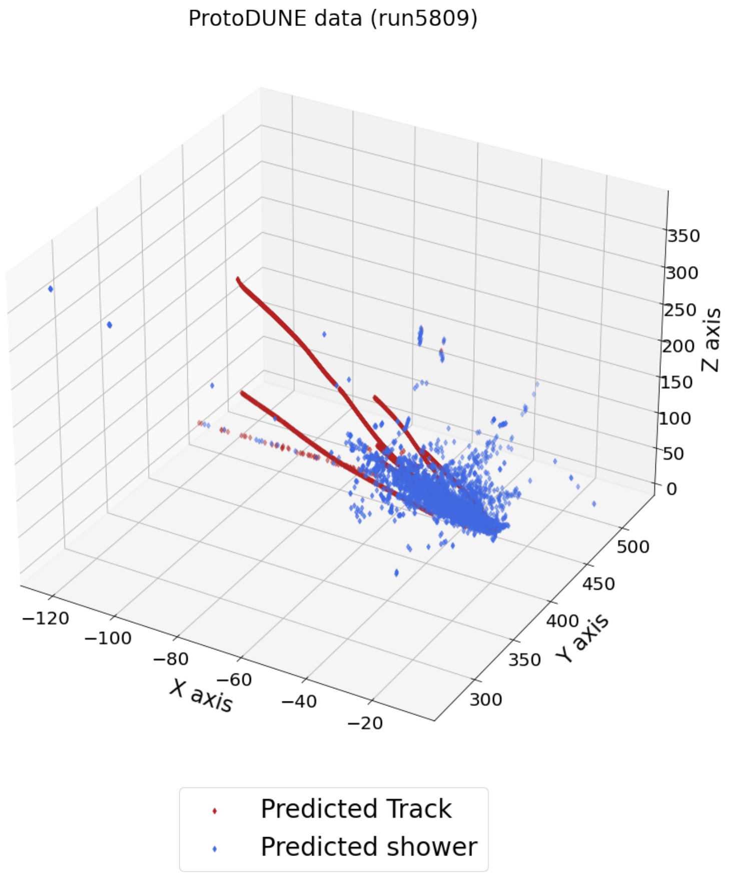


Figure 4.16: SparseNet when applied on a ProtoDUNE-SP dataset from Run 5809.

classification task and it has been evaluated in detail on both MC and ProtoDUNE-SP data. An analysis was also performed to understand the response of the SparseNet on different particle types.

Results for both the track and shower class show purities and efficiencies greater than 90% when using the SparseNet on data collected by ProtoDUNE-SP during Run 1. The accuracy metric for track and shower hits also shows promising results, outperforming the currently adopted algorithm in ProtoDUNE-SP. Therefore, such novel classification algorithm represents a promising solution for the larger DUNE-FD detector modules. Finally, the SparseNet has been integrated into the ProtoDUNE-SP analyzer framework and further tests are needed to completely evaluate the performance of the model.

Summary of contributions:

- Feature extraction: selection of the most relevant quantities for discriminating between track and shower hits
- Model testing and validation
- Tuning of the classification threshold
- Development of particle-based performance metrics for the network evaluation
- Evaluation of the SparseNet model on MC and ProtoDUNE-SP data
- Integration of the SparseNet model into the ProtoDUNE Analyzer software package

5

Data acquisition system of the DUNE experiment

The scope of this chapter is to introduce the design and the architecture of the data acquisition system of the DUNE experiment. This chapter aims at introducing the relevant elements of the data acquisition system, covering in particular the readout, storage and dataflow aspects which represent the main topic of this research work. In terms of data acquisition components, more details are given to the readout system of both the ProtoDUNE DAQ and the DUNE DAQ which are further discussed in chapter 7. Note that most of the material in this chapter has been taken from the DUNE technical design report for the single phase detector [9] and the Trigger and Data Acquisition System Specifications [18] report for the DUNE experiment.

5.1 The DUNE trigger and DAQ system

The role of the DUNE-FD trigger and data acquisition (DAQ) system is to receive, process, filter and store the data produced from the different detectors. The data acquisition system is designed so that all Far Detector modules will use the same design. The individual modules are serviced independently and loosely coupled together through a cross-module triggering system.

The DUNE DAQ system will be hosted into two sites: the underground central utility cavern

(CUC) and the control room at the Sanford Underground Research Facility (SURF). The former is responsible for the interface with the detector and some pre-processing, while the latter is responsible for the buffering, monitoring and the run control.

Overall, the DAQ is composed of the Readout responsible for receiving data from detectors, buffering them and extracting features, so called trigger primitives. A Data Selection (DS) system is then responsible for selecting the most interesting information by using the trigger primitives. This leads to a trigger decision command which is forwarded to the Dataflow (DF) system whose task is to orchestrate the flow of data from the memory buffers of the Readout to a persistent storage system (Output Storage). A further selection is also performed by the Trigger to keep only the most relevant physics data. The final step is transferring to the offline storage, located inside the Fermilab computing center. Figure 5.1 shows the conceptual design of the DUNE-FD data acquisition architecture for a single 10 kton module. The figure also highlights which components of the system are hosted in the underground facility and which ones inside the on-surface control room. In terms of distances, the Central Utility Cavern is located right across the detector, wehereas the On-surface Control Room is 1.5 km from the detector cavern.

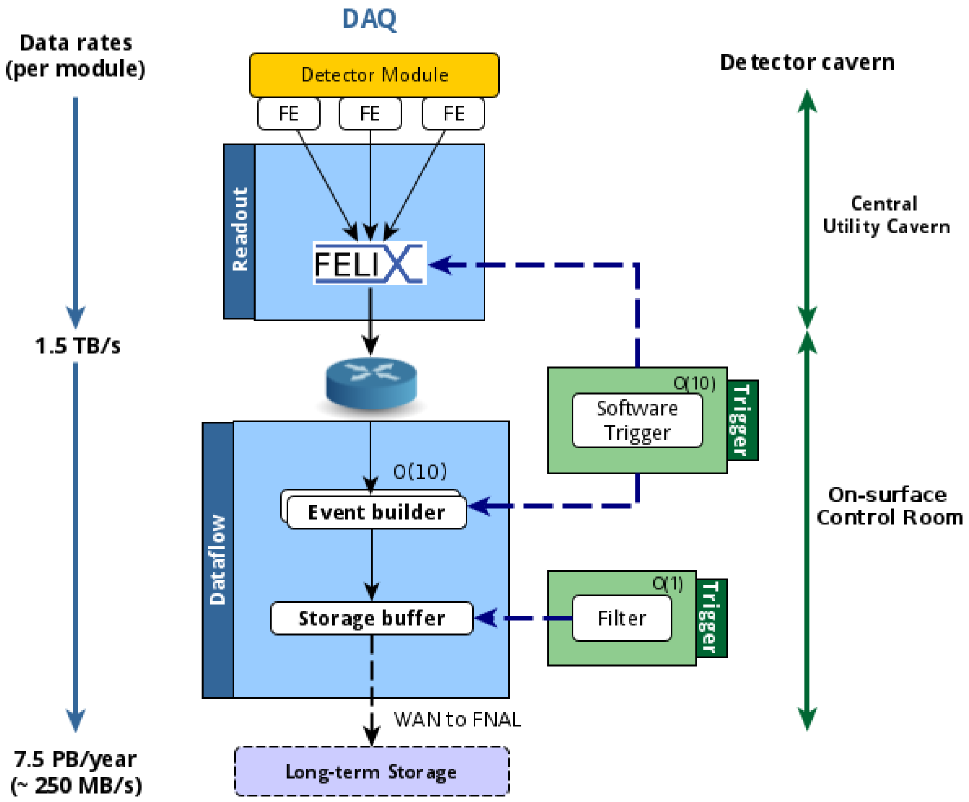


Figure 5.1: High-level conceptual design of the DUNE-FD data acquisition system for a single 10 kton module.

5.1.1 Physics constraints on the DAQ

The physics goals of the DUNE experiment heavily drive the DAQ design. The experiment's DAQ system must support the data-taking of events relevant to the wide physics program: measuring neutrino oscillations from the beam from Fermilab; measuring the properties of atmospheric neutrinos; searching for baryon-number violating processes; studying neutrino burst events from core-collapse supernovae.

From the detector perspective, the DAQ system processes the data originating from mainly two different sources: the time projection chamber (TPC) and the photon-detectors (PD). In the case of the TPC, the signals are produced by ionization charge measurements and require a data recording over a time window between 1 ms and 10 ms. This is due to the electron drift speed in liquid argon and the size of the detectors. Typically, the drift time value is set to 4.25 ms by taking into account a drift electric field of minimum 450 V/cm as well as a factor for the production of the ionization electrons. For the DUNE detector, the maximum time window is set to be at least twice the drift speed, hence the 10 ms of readout window. Further details can be found in the technical design report of the experiment [9]. For the case of the PD system the detection principle is based on the scintillation light emitted in liquid argon which usually happens over a timescale between 1 ns and 1 μs. Thus PD signals need high sampling rates (62.6 MHz) compared to the TPC data that are sampled and digitized at a slower rate (2 MHz). However, by applying zero-suppression and by taking into account the expected production rates the contribution from the PD has a significantly lower impact compared to the TPC data.

The large physics program of the DUNE experiment results in the study of many different types of events. The expected activity rates as a function of the signal energy for a single 10 kton detector is shown in figure 5.2. As it can be noted, the most significant activity originates from low energy (less than 10 MeV) events, typically due to the radiological background of the detector. There is also a minor contribution to the rate at low energy due to the solar neutrino interactions. Supernovae burst (SNB) events typically span an energy range between 10 MeV and 30 MeV with an expected rate in the galaxy of one per century [2]. At energies higher than 100 MeV the activity is dominated by beam, atmospheric and cosmic ray interactions. The consequence on the DAQ system is that the signals associated to the several physics activities are localized in both space and time across the detectors. This is valid for most of the interactions at the exception of SNB events which, typically, have a signature characterized by many low-energy neutrino interactions over an extended period of time.

Having signature rates spanning different orders of magnitude means that the DAQ data selection system has to handle mainly two different scenarios: short localized high-energy interactions and many low-energy extended activities. The first case is typically associated with events whose energy is higher than 100 MeV while the second case corresponds to energy deposits of approximately 10 MeV. The DUNE data selection system has been designed to be deployed in a hierarchical way to provide, at each triggering level, enough processing power and time to form the data selection decision.

The DAQ triggering system will need to be able to cope with multiple constraints driven by the DUNE physics mission. A self-triggering mechanism is needed to identify the many low energy deposits for supernovae burst events. In addition, the data taking must be carried out with a very high detector uptime since the some of the target events (e.g. SNB events) are rare and require the collection of sufficient data to provide a statistically relevant sample.

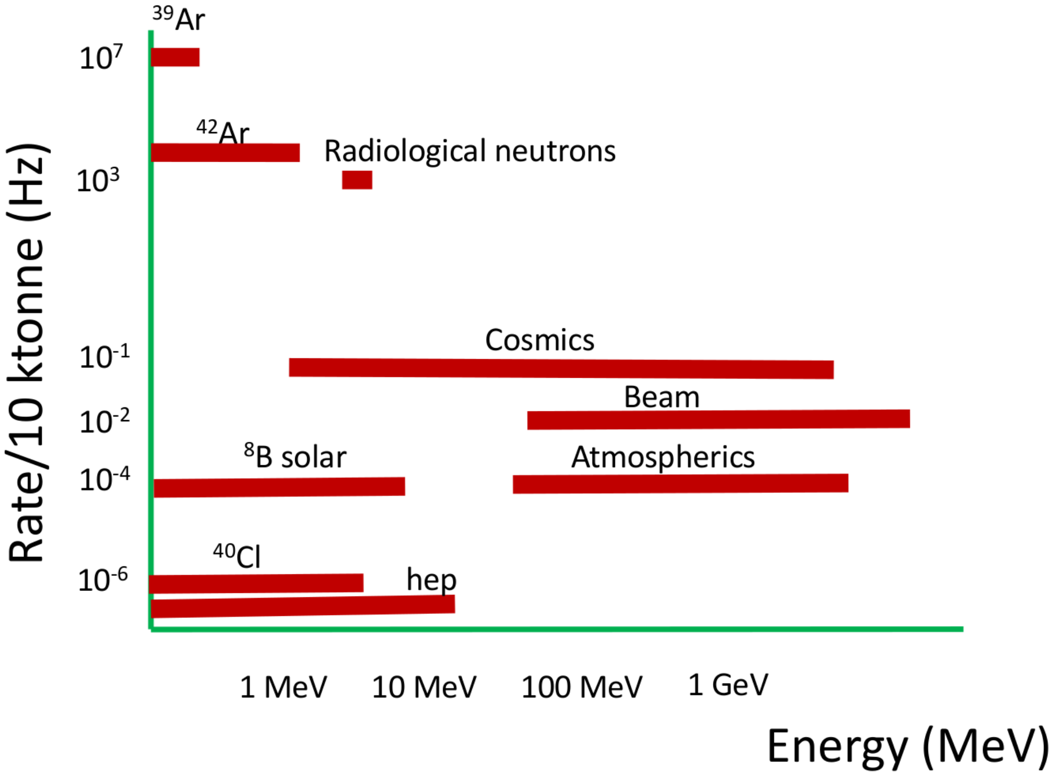


Figure 5.2: Expected event rates as a function of the energy for a single FD module. Figure taken from [9].

5.1.2 DAQ System Design

The DUNE DAQ is composed by many different subsystems: Readout, Data Selection (DS), Dataflow system (DF), DAQ Timing and synchronization, Output Storage (OS) and Control, Configuration and Monitoring (CCM). Figure 5.3 shows an overview of the DAQ architecture for a FD module. It is also worth noting that there are also many other external components (depicted in grey) that the DAQ needs to interface with: these include the TPC Cold Electronics, the Photon Detector readout system, the offline computing infrastructure, the cryogenic instrumentation, the slow control and the calibration system. The different subsystems must interact in a coordinated manner for successful data taking and to reduce possible failures. In general, the role of the DAQ system begins with the raw digital information from the detector electronics and it concludes with the network transfer of the collected data to the offline computing centers.

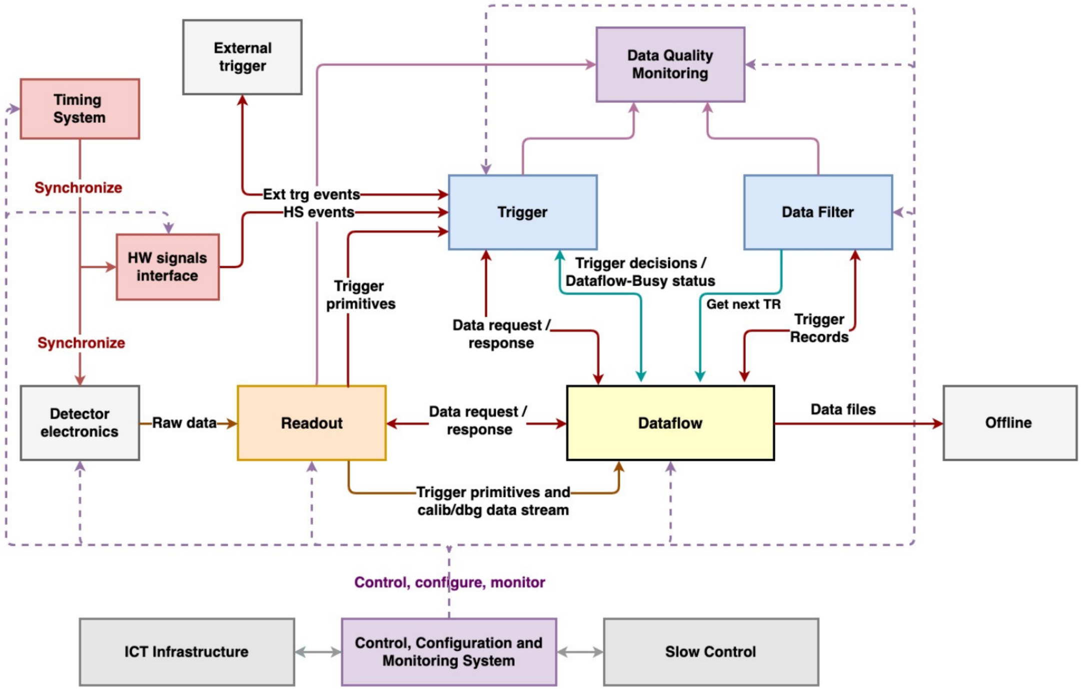


Figure 5.3: General overview of the DAQ architecture of a FD module. External systems are depicted in gray, whereas the remaining elements are either software and hardware components. Figure taken from [9].

5.1.3 Readout

The Readout (also known as *Upstream DAQ*) is the closest system to the detector electronics. Its task is to interface the detector front-end with the DAQ processing units. This is accomplished in both hardware and software. Figure 5.4 shows a conceptual overview of the Readout subsystem. This is composed of many readout units (RUs) that act as a data receiver, a buffer, and a low-level data selection system. Each readout unit consists of two high-performance input/output (or I/O) devices and storage hardware equipped in a commercial off-the-shelf (COTS) server. The readout unit is physically connected to the detector electronics with optical fibers and serves the data to the other DAQ subsystems via a switched network.

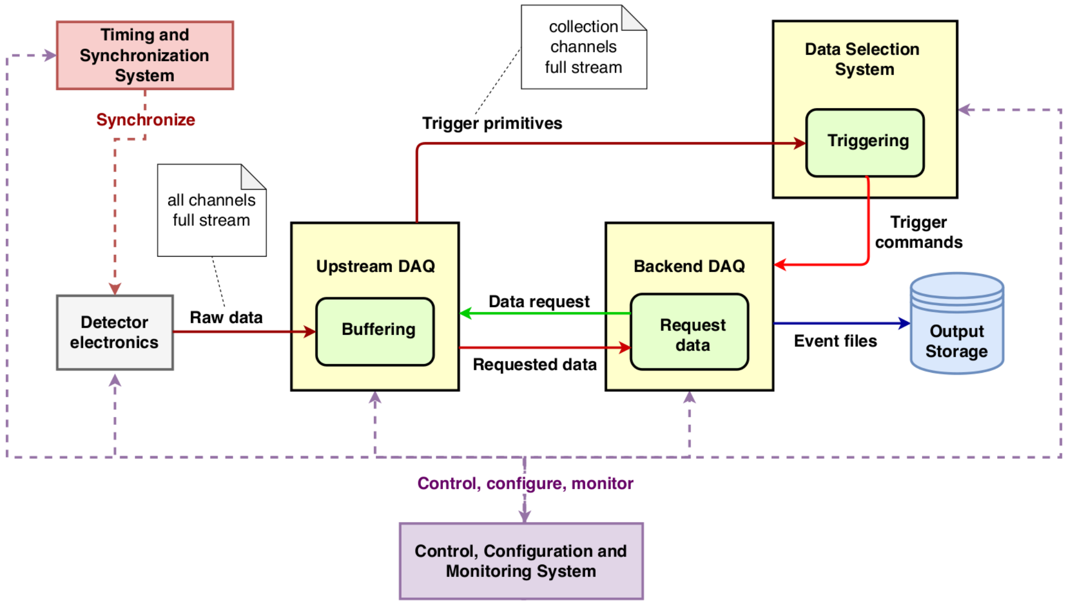


Figure 5.4: Conceptual overview of the Readout system. Figure taken from [9].

The modular nature of the DUNE apparatus enables splitting the Readout system into 150 identical I/O devices (per detector module) operating in parallel. A single I/O device is used to readout a single APA. In the current baseline design the plan is to have 75 readout units to extract all the data from TPC detector and 8 for the PD system.

The high-performance I/O device planned for the DUNE readout is a general-purpose device, initially developed by the ATLAS collaboration and known as FrontEnd Link eXchange (FELIX) system [19]. The FELIX system was developed as a generic solution for routing custom serial detector links from front-end ASICs and FPGAs to data collection and processing components via a commodity switched network. Figure 5.5 shows an image of a FELIX-712 PCIe card used at present, highlighting the main components of the system. In practice, a FELIX device is a shared software/firmware solution hosted on a COTS server. An FPGA-based PCIe I/O card is used to stream the input data from the detector front-ends into a circular memory buffer in the host server using continuous DMA transfer. Finally, the host server runs software to route the data to multiple output destinations (several network clients).

There are several tasks that the Readout needs to perform. These are summarized below:

- **Detector interface:** the FELIX readout cards are designed to handle an optical link bandwidth of approximately 10 GB/s. The front-end electronics of each TPC anode-plane

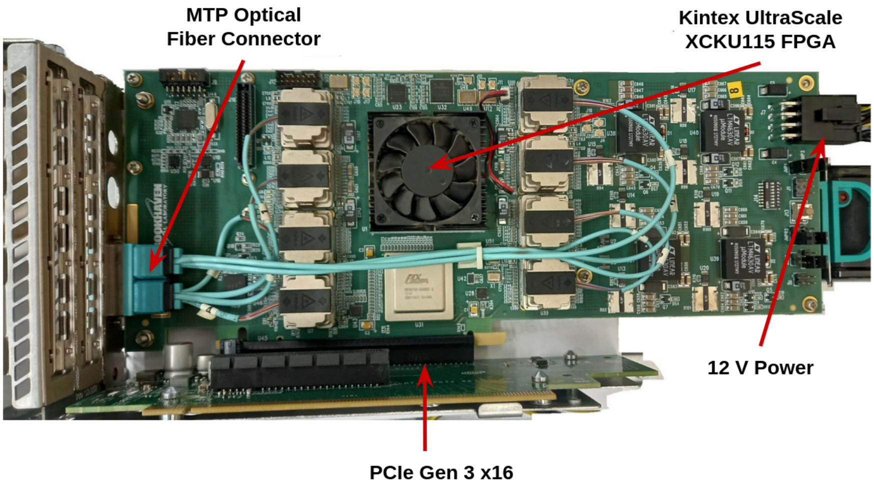


Figure 5.5: Image of FELIX-712 PCIe card used at present.

assembly are, in fact, connected to the card through ten links of 1 GB/s using the 8b/10b encoding scheme. Note that the FELIX board, firmware and software were initially designed for the Phase-I upgrade of the ATLAS experiment. However, the functionality of the FELIX technology has been adapted in the context of the DUNE experiment and has been extensively tested in ProtoDUNE [19].

- **Primitive generation:** the firmware of the readout system is also designed to perform primitive generation for both the TPC and PD subsystems. This is accomplished by a real-time reorganization of the incoming data and by applying noise filtering algorithms. The result of this initial processing is a trigger primitive (TP) which describes time periods in which the waveforms are noise-free. The Data Selection system then uses trigger primitives to form a trigger decision to keep the data from a specific regions of interest.

- **Local storage buffer:** the Readout acts also as a buffering stage for the detector data. This ensures that the information is safely stored while waiting for a trigger decision from the DS system or while waiting for the transfer to the DF system. It is crucial to make sure that size of the buffer is sufficiently large to allow both localized and extended triggers. Extended triggers are primarily used to record Supernova Burst Events.

Incoming data must be real-time processed and temporarily buffered for several seconds in the readout system in order to extract the interesting data regions for trigger decisions. One particular decision (Supernova Burst event or SNB) includes the permanent storage of the full complement of the raw data stream for over 100 seconds. This is because the SNB events require data access to the information stored on a longer time scale as compared to more localized events. More details on the implementation and characteristics of readout system for SNB events will be described in detail in chapter 7.

5.1.4 Data selection

The Data Selection (DS) or Trigger System is mainly a software system designed to filter the data collected by the DUNE detectors (TPC and PD). The DS provides a trigger decision sent to the other DAQ subsystems in the form of a trigger command. For example, when the Readout receives the trigger signal, it will then send the collected data to the Dataflow System.

The tight physics requirements of the DAQ have significant consequences on the Data Selection system. In fact, a very high selection efficiency (>95%) needs to be sustained for all types of neutrino interactions (beam, atmospheric, radiological processes, etc.). In addition, the DS also needs to reduce the data rates to a maximum output of 30 PB/year to the offline computing center. Therefore, trigger algorithms have to be tuned accordingly.

The trigger decision is formed in a hierarchical scheme as illustrated in figure 5.6. First, low-level decisions are computed based on hits on the single channels. These are known as *Trigger Primitives* and are computed within the Readout in an FPGA. In a second step, a high level decision is formed (*Trigger Candidates*) by analyzing clusters of hits. The final step is the trigger command which can be a localized activity in the detector or, in the case of an SNB interaction, it can be an extended low-energy event across the whole detector volume. The recorded events from a trigger command are subject to a final data filtering selection before being sent out to the offline storage system.

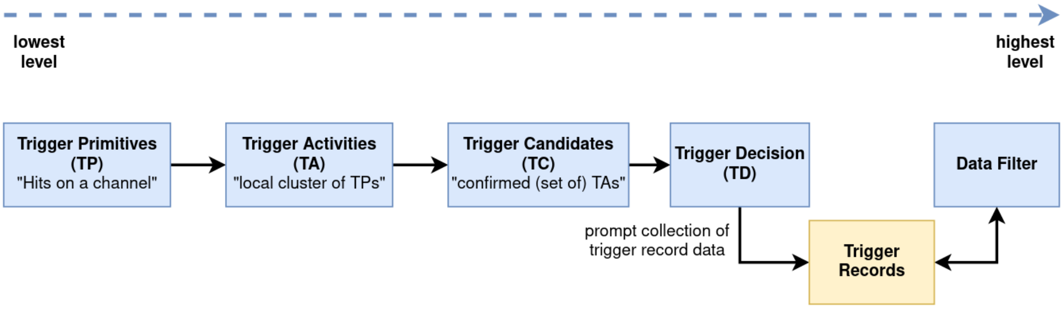

lowest level
highest level
Trigger Primitives (TP) "Hits on a channel"
Trigger Activities (TA) "local cluster of TPs"
Trigger Candidates (TC) "confirmed (set of) TAs"
Trigger Decision (TD)
Data Filter
prompt collection of trigger record data
Trigger Records

Figure 5.6: Conceptual overview of the DUNE data selection strategy. Figure taken from [9].

5.1.5 Dataflow system

The responsibility of the Dataflow System (DF) is to move the data selected by the DS from the Readout to the Output Storage system. In general, the DF accepts trigger commands and it forwards the query to the relevant buffers in the readout system in order to receive the selected data entries.

The DF system has two main components: the Data Flow Orchestrator (DFO) and the Event Builder (EB). The role of the DFO is to accept the trigger commands and forward their execution to many EB processes. This is done to make sure that the target rate is achieved and provide redundancy to the system. In addition, the DFO is also responsible for collecting operational metrics (status, the health of the system, capacity of the buffers, etc.) of all the DF processes to be able to promptly act on them in case of failures. The EB processes are responsible for querying the appropriate Readout nodes, requesting the data as indicated from the DS system. The last step is to process and aggregate the chunks before sending them to the Output storage system.

5.1.6 Output storage system

The Output storage system is a large storage buffer whose task is to receive the filtered data from the DF system before transferring them to the Fermilab offline computing center. The system is designed to provide a storage buffer of one week of data taking, approximately 1 PB. This is the result of a trade-off between the required I/O rate and the storage needed in case of problems in both the DAQ system as well as in the optical link connection with Fermilab. Finally, the Output storage will also send the data flow status and other relevant statistics to the operational monitoring system.

5.1.7 Control, configuration and monitoring

The control, configuration and monitoring (CCM) software of the DUNE DAQ manages both the DAQ system and the components of the detector that participate to the data-taking. The CCM is responsible for many tasks:

- Provide a central access point for controlling the DAQ components in a hierarchical manner. Steer the data acquisition in a coherent way (Run Control)
- Configure the DAQ components and the detector front-end electronics by providing a

description of system components, ability to modify configurations and graphical user interfaces for configuration accesses

- Provide error handling and recovery mechanisms to provide a smooth, robust and fault-tolerant data-taking

- Provide operational monitoring and message logging during data-taking periods. This is achieved by aggregating in a scalable manner the messages originating from the detector sub-components and the DAQ processes

All of the tasks performed by the CCM have the main goal of maximizing the overall system up-time, keep a high data-taking efficiency and ensure good data quality standards.

5.1.8 Experimental challenges for the DAQ system

By design the DAQ architecture is a scalable system. It is designed for a single DUNE-FD module but it can serve all the other detectors that will be installed over time. In addition, the DAQ's goal is also to record and store all the data produced with a very high data-taking efficiency and with zero dead time. This creates constraints on the filtering, data selection and compression. Finally, it is also worth noting that the DAQ system will also change with time. For example, the design is very conservative on the amount of data transferred and recorded but these will be relaxed with more experience from both the detector operation and the physics analysis. This approach is very similar to the one adopted for the LHC experiments during their first years of operation.

There are many challenges that the DUNE DAQ needs to deal with. Having a very uptime (larger than 95%) requires the system to be redundant and reliable. As a consequence, the DAQ system needs to be fully controllable and configurable without the need to be based in a control room. This ensures that the overall downtime is kept as low as possible. As a comparison, the typical uptime of collider experiments is in the orders of 30% because of the continuous interventions, upgrades or accelerator cycles. Finally, remote monitoring of the operational data taking parameters (e.g. data rates, buffer sizes, storage capacity, network throughput, etc.) and automated recovery mechanisms will also be put in place to make the system fault tolerant. This is motivated by the need to minimize errors during data taking and, therefore, ensure high data-taking efficiency. Considering the wealth of physics sources the DAQ system must be very robust and flexible. This is because the wide range of readout windows and trigger rates require the system to cope with large events (SNB) and localized (space and time) events.

It is important to mention that operating a large and complex detector underground also creates many challenges for the DAQ system. The infrastructure's installation, operation, and maintenance will be heavily affected by the limited accessibility to the site and the lack of available power and cooling. This means that, whenever it is possible, it is best to move the hardware components at the surface site. For this reason, the Dataflow machines, the large storage buffer and the event filtering systems will be placed inside the control room on the surface of the DUNE site.

5.2 ProtoDUNE DAQ: readout system

The ProtoDUNE-SP DAQ relies heavily on the design of the DUNE-FD DAQ. The software and hardware used in ProtoDUNE-SP has been in constant evolution to prepare the system for the DUNE experiment. The idea behind the DUNE-DAQ software is to be able to develop and deploy the software for the DUNE data acquisition system and test it (with some minor modifications) on the ProtoDUNE-SP setup.

Front-end Electronics

As outlined in chapter 3, the ProtoDUNE-SP TPC consisted of six Anode Plane Assemblies each with 2560 wires and readout with 20 Front End Mother Boards (FEMBs). The two main data producers of ProtoDUNE-SP are the TPC and the PD subsystems. The TPC wire planes and the photon detectors are read out via digitization electronics that are placed on top of the cryostat. Specifically, the TPC is read out via Warm Interface Boards (WIBs) that assemble the charge data from the TPC wires originating from four FEMBs into a fixed-sized packet to the DAQ. Therefore, a single APA is readout via five WIBs and a total of 30 WIBs are used for the ProtoDUNE-SP detector.

The photon detector is read out via Silicon Photo-multiplier Signal Processor modules (SSPs). SSPs contain a waveform digitizer and a discriminator that is used to trigger on light signals. In addition, four SSPs are used to read out the data originating from a single APA, totaling in 24 SSPs for the whole detector. Both the WIBs and the SSPs are coupled with optical connections to and from the DAQ system to minimize the electrical noise on the front-end electronics.

5.2.1 TPC readout and data volume

In the ProtoDUNE-SP detector, the APAs of the TPC are readout with the FELIX I/O card. The 5 WIBs of a single APA send data at 2 MHz frame rate per optical link to the FELIX readout. Each WIB multiplexes the data to two lines of 9.6 Gb/s. Therefore, 10 optical links are needed to fully read out a single APA with the FELIX system. Additionally, each WIB frame is accompanied by a CRC20 checksum (generated by the WIB firmware) that is evaluated by the FELIX firmware to verify the integrity of the data. In case of CRC errors, the recorded data are marked with an error flag. Note that each WIB frame contains 120 32-bit words (size of WIB frame without encoding is 464 bytes), leading to a total payload data rate of approximately 7.68 Gb/s. Combining the payload data rate with the 8b/10b encoding and the transmission protocol headers, the total line rate for a single WIB link reaches the nominal bandwidth of 9.6 Gb/s.

5.2.2 FELIX hardware and firmware

The FELIX I/O interface card used in the ProtoDUNE setup is the FELIX hardware platform (FLX-712) developed by the ATLAS collaboration. This consists of a PCIe Gen3 card based on a Xilinx UltraScale FPGA (XVKU115) capable of sustaining 48 bidirectional high-speed optical links via MiniPOD transceivers installed on the device (see figure 5.5).

The readout node hosting the FELIX I/O card is subjected to a challenging load that requires a lot of processing power (load on the host CPU) to sustain the high rate of incoming frames (2 MHz) and the high payload data rate (74 Gb/s). Therefore, modifications to the original FELIX firmware were introduced to reduce the overall flow of data. This is done by grouping the incoming frames to minimize the processing operations (e.g. memory copies) and the total number of network calls (e.g network I/O operations). The resulting aggregated frames are sent as a single network message. In addition, the size of the DMA transfer between the FPGA and the host memory has been tuned to allow the optimization of frame parsing (reduce as much as possible potential split data blocks) and to ease the serialization of data for network transmission. Note that by increasing the grouping factor of the WIB frames and the DMA payload size, the total rate of operations decreases at the cost of an increased time needed from the host node for memory access. The system has been tuned with a grouping factor of 12 WIB frames and a DMA payload size of 4 KB. Therefore, the rate of each link transferring data from the WIBs to the FELIX system is: WIB size of 5568 bytes (superchunk block size) at a rate of

166 kHz (exact throughput of 881 MiB/s[1]).

5.3 DUNE-DAQ software and readout system

The DUNE-DAQ software is the collection of applications, plugins and packages that, at the time of writing, is being developed to support the DUNE data acquisition system. Multiple challenges had to be solved when designing the DUNE-DAQ software. For what concerns the Readout system, the main challenges are:

- Wide number of front-end electronics components (e.g. TPC electronics, silicon photomultipliers, etc.)
- Support many I/O devices: the Readout system is made of COTS servers that are equipped with PCIe FPGA boards (FELIX system), network interface cards (NICs), storage devices (e.g. SSDs)
- Support different combinations of arrival rate and payload sizes
- Quasi real-time performance: high-throughput (approximately 10 GB/s) I/O cards, provide a low latency response and scalability to hundreds of servers

The design of the DUNE-DAQ Readout system has followed a number of key requirements. First of all, the Readout system needs to support all possible front-end types which also may have different data rates and payload sizes. In addition, data must be buffered for a certain amount of time that can last from a few seconds to a few minutes for debugging purposes or because of special data selection requests (e.g. supernova neutrino burst trigger). Another requirement is to be able to respond to data requests with time-windows that may very from microseconds to seconds. Therefore, the system must be able to provide data indexing to ease the search for the requests. Finally, the Readout system also needs to be perform data processing on the incoming data for error and consistency checks or for performing custom algorithms for data selection (e.g. hit-finding).

Based on these requirements, the Readout system has been designed by keeping the core functionalities fully generic to support all the data taking needs of the experiment and to avoid code duplication. Figure 5.7 illustrates the data-flow diagram of the design of the Readout system. As it can be noted, the Readout system is divided into four main domains: front-end, data pro-

[1] The *mebibyte* (MiB) is a unit of computer data storage. It is equal to 1024 KiB or 2 to the power of 20 bytes.

cessing, buffering and storage, data request/response. The front-end domain is responsible for coupling the DAQ system with the detector electronics whereas the data processing domain performs a combination of pipelines, both pre-processing and post-processing, on every raw data frame (e.g. check for error, check for debug/calibration flags or hit-finding feature extraction). The buffering and storage domain (also known as latency buffer) is responsible for storing the data and for providing a lookup routine based on a unique identifier (e.g. timestamp). The latency buffer also provides custom memory allocation policies such as NUMA-aware allocation[2] or data aligned allocation. Finally, the Readout system needs to respond to data request (data request domain) from other DAQ subsystems (e.g. event builder, data quality monitoring). The default approach in this case is to copy the data from the latency buffer and send it to the target application. A periodic cleanup routine is also activated to remove the non-requested data.

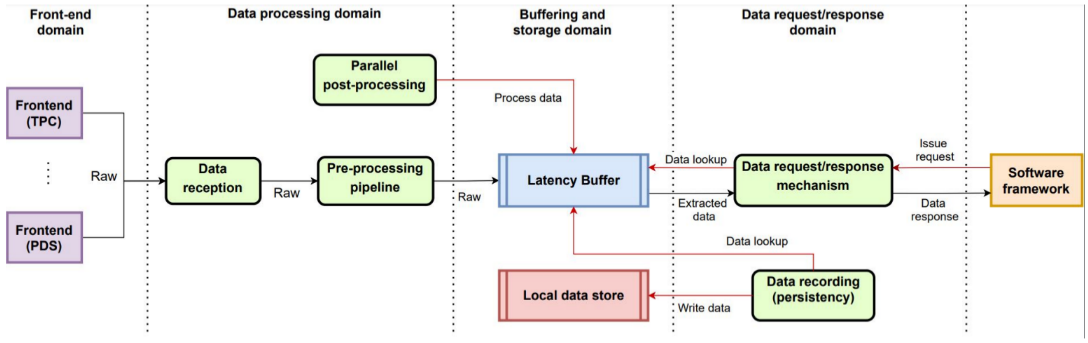


Figure 5.7: Data-flow diagram of the design of the Readout system within the DUNE data acquisition system. Figure taken from [18].

[2] NUMA (Non-Uniform-Memory-Access) is the method of configuring a multi-socket architecture (e.g. dual socket server) such that multiple processes can share memory locally.

6

ATLAS TDAQ system for the Phase-II upgrade

The objective of this chapter is to introduce the relevant elements of the data acquisition system for the Phase-II upgrade of the ATLAS experiment [1] which is expected to start taking data from the late 2020s. Compared to chapter 5 where a greater level of detail is given to the DUNE detector, the ATLAS data acquisition system is described only in view of the Phase-II upgrade focusing in particular on the storage and the dataflow components which will be relevant for chapter 8. The evolution of the TDAQ architecture is also discussed in chapter 8. Similarities and differences between the data acquisition systems of both the DUNE and ATLAS detectors are also illustrated. Most of the material for this chapter is taken from the ATLAS technical design report for the Phase-II upgrade of the TDAQ system [20].

6.1 Introduction

The ATLAS experiment is going to upgrade the detectors and the trigger and data-acquisition (TDAQ) system in order to take advantage of the full potential provided by the HL-LHC upgrade expected to start producing collisions in 2028. In particular, for the Phase-II upgrade the TDAQ system will be completely redesigned as a consequence of the harsher environment and upgraded detectors. The initial baseline architecture foreseen for the Phase-II upgrade consists of a single-level hardware trigger operating at 1 MHz within a fixed latency of 10 μs. This is followed by the Readout system which will receive front-end electronics data at the L0-trigger rate (1 MHz) before sending them to the Dataflow system for an estimated data traffic of 5.2 TB/s. The task of the Dataflow system is to buffer, transport and format the event data. It acts as an interface between the Readout and the Event Filter systems. Once the data are buffered in the Dataflow (DF) system a more refined selection is executed by the Event Filter (EF) system. The EF is processing farm of commodity servers whose task is to select the most interesting events to send to the Dataflow system. It is expected that the total request bandwidth (i.e. read throughput) from the Dataflow to the EF is 2.6 TB/s. Once events are accepted and formatted, they are then sent to permanent storage at a throughput of approximately 50 GB/s (average rate of 10 kHz). Figure 6.1 shows a schematic representation of the main functional blocks of the DAQ architecture for Phase-II [20].

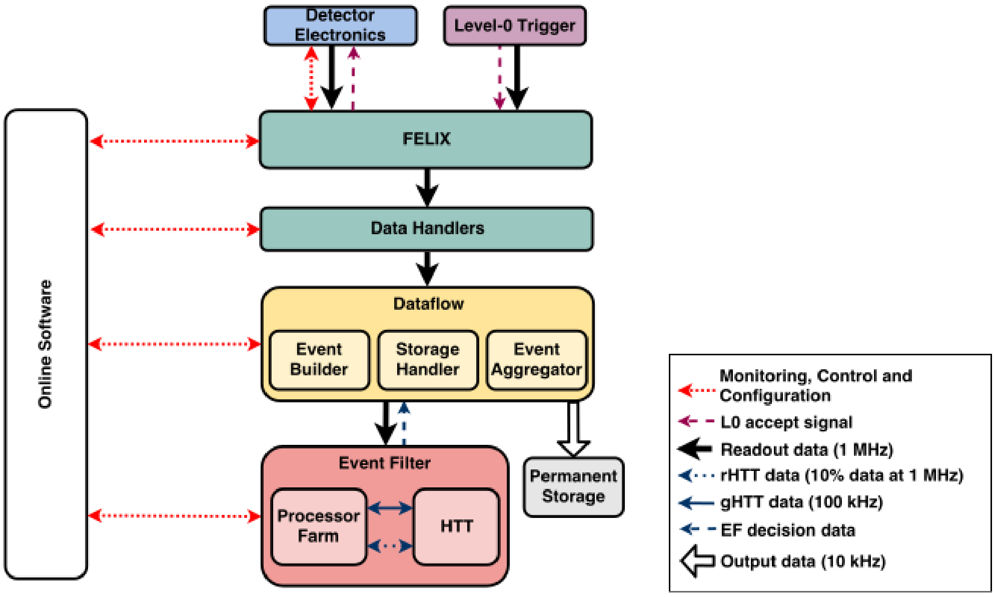


Figure 6.1: Schematic representation of the DAQ architecture for Phase-II. Figure taken from [20].

6.2 Readout system

The Readout system receives data from the ATLAS detector front-end electronics (FE) at a rate of 1 MHz. This corresponds to the output rate from the first level of triggering (L0 trigger) which filters incoming data from the initial 40 MHz collision rate. The Readout system also performs basic processing before transmitting data to the Dataflow system. The detector data are sent via a custom serial link to the Front-End Link eXchange (FELIX) subsystem which provides a common interface to the custom detector front-ends. This will need to sustain a total bandwidth of 5.2 TB/s at 1 MHz. The data is then sent to the Data Handler (DH) nodes via a dedicated network. The task of the DH is to perform detector specific formatting, providing a monitoring interface to the Readout system and routing the data to the Dataflow system.

From an implementation point of view the FELIX system will be built on top of approximately 300 commodity servers with custom FPGA boards and with more than 15000 links connected. The Data Handler will be implemented on more than 500 servers each connected to both the Readout network and the Dataflow network. On average the output fragment size from each Data Handler is approximately 10 KiB.

6.3 Dataflow system

The Dataflow system is a key element of the ATLAS DAQ architecture. It is divided into three components: Event Builder, Storage Handler and Event Aggregator.

6.3.1 Event Builder

The Event Builder (EB) is the logical interface between the Readout and the Dataflow system. It is responsible for mapping the event data with the corresponding accept message originating from the L0 trigger. This information will later be used by the Event Filter farm to perform a more refined selection. The Event Builder is also in charge of overlooking the data movement across the whole system and for communicating the back-pressure from the Dataflow to the Readout systems.

6.3.2 Storage Handler

The Storage Handler is a large buffering system which will need to sustain an aggregated I/O throughput of 7.8 TB/s. This corresponds to 5.2 TB/s of data input from the Data Handlers and 2.6 TB/s of data output to the Event Filter farm. The Storage Handler will also hold the data of the Event Aggregator at a rate of approximately 50 GB/s, negligible to the total I/O of the system. The main advantage of having a large storage buffer is to decouple the Readout from the Event Filter. This, in turn, will allow a trade-off between processing power and storage resources. In fact, it will be possible to take advantage of the time used to refill with proton bunches the LHC accelerator (inter-fill time between data taking sessions or runs) and process some of the recorded events. In this way it is possible to make sure that the Event Filter farm is always being utilized. Moreover, the large buffering system can also provide robustness in the Event Filter farm in case of varying operational parameters due, for example, to changes in the trigger configuration.

6.3.3 Event Aggregator

Another component of the Dataflow system is the Event Aggregator. This is the system that formats, groups and compresses the selected events from the EF before sending them to permanent storage at an output rate of approximately 50 GB/s. The Event Aggregator shares the storage hardware with the Storage Handler system and, thus, it acts as a logical component of the Dataflow system. As required by CERN-IT department, the Event Aggregator needs to provide buffering capabilities for up to 48 hours, leading to a buffering system of approximately 8 PB. As for the current architecture, this buffering system will decouple the online data taking from the offline systems which in turn will make the design more robust in case of disruptions on the data transfer to the IT permanent storage. Other tasks of the Event Aggregator consist in ensuring that event data is sent to Tier-0 with the proper output stream.

6.4 Event Filter

The last component of the data acquisition chain before storing the data in the Event Aggregator is the Event Filter (EF) selection. The EF system takes as input the data from the Storage Handler system and it selects the most interesting events according to a menu-driven event selection. The EF is a multi-threaded processing farm of commodity servers (approximately 3000 nodes) that will produce, after selection, an average output throughput of 50 GB/s. This corresponds to an average rate of approximately 10 kHz which is a reduction by a factor 100 with respect to the previously selected data events stored in the Storage Handler system.

6.5 Implementation of a sliced system

From an implementation perspective, the Dataflow system is designed in a sliced structure with several independent networks. Considering the number of detector readout channels and the available ports on the switches it is foreseen to have approximately 35 slices which will all be interconnected by one core routing system. This will also provide the interconnection to the EF farm. Therefore, multiple DH nodes are connected to the SH nodes which, in turn, are accessible by the Event Filter via central routers. Figure 6.2 shows a diagram of the ATLAS Phase-II

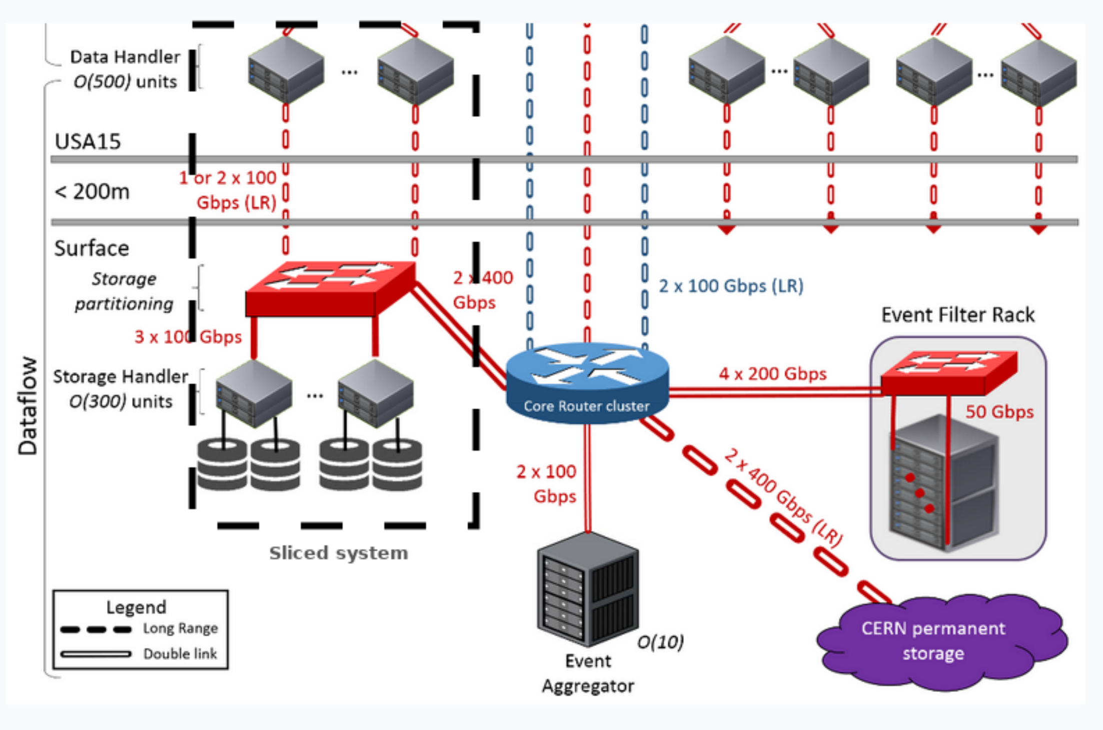


Figure 6.2: Diagram illustrating the ATLAS Phase-II architecture. Figure taken from [20].

architecture highlighting in particular a single Dataflow slice. The figure also illustrates the network interconnection speeds between the data acquisition components: Data Handler, Storage Handler, Event Aggregator, Event Filter farm or CERN permanent storage. In addition, figure 6.2 shows that only the Data Handler nodes will be located underground inside the ATLAS cavern. This is because of the latency constraints due to the high input data rates. Therefore, the Data Handler servers need to be close to the detector front-end electronics. The remaining computing farm (transient storage, Event Filter farm, etc.) will be located on the surface of the ATLAS site.

The Dataflow design originates from the network topology. In fact, the system is divided into two networks: one for the sliced networks and one for the global aggregation of the events from the Storage Handlers to the EF farm. The whole Dataflow system has been designed to accommodate only once data transfer through the router: data will be sent from the Dataflow system to the EF once and upon the decision of the trigger only metadata information is sent to the router for the event building or for rejecting events. In this way, it is possible to reduce the total throughput that the Core Router cluster will need to sustain. Finally, the advantage of having a sliced architecture is the modularity that comes with its design. In fact, it will be possible to include more slices in case additional bandwidth will be needed.

6.6 System sizing

To summarize the constraints, the Dataflow system and, in particular, the Storage Handler will need to be able to sustain at 1 MHz an input data rate of 5.2 TB/s and an output rate of approximately 2.6 TB/s, which corresponds to a total aggregated I/O throughput of 7.8 TB/s. In order to implement such a system approximately 1800 NAND-based solid state drives (SSD) will be needed. This is based on the assumption that the storage devices are based on the fourth generation of the PCIe technology and, therefore, a total write bandwidth of 5 GB/s will be available. Therefore:

$$\text{Number of SSDs} = \frac{7.8\ \text{TB/s}}{5\ \text{GB/s}} \approx 1800\ (+15\ \%\ \text{margin included})$$

In terms of sizing, the total input throughput is given (write clients) by approximately 500 readout nodes whereas the storage system is implemented on a pool of approximately 300 nodes. The system must be able to serve 3000 nodes from the Event Filter farm (read clients). Therefore, each Storage Handler node will host 6 SSD drives. In terms of throughput, this corresponds to roughly 20 GB/s for writing and 1 GB/s for reading per Storage Handler node. In addition, assuming a 10 TB SSD drive, the total transient storage system size is 18 PB. Such a storage system can buffer, at the ATLAS Phase-II write throughput, more than one hour of data.

6.7 Comparison between the DUNE and ATLAS DAQ systems

Although the goals of the DUNE and ATLAS detectors are very different, the data acquisition system of the two experiments has some similarities. First of all, the total data throughput that both experiments will need to sustain is a few TB/s. Both experiments, in fact, make use of a PCIe-based readout (FELIX system) to extract the data from the optical fiber links originating from the front-end electronics. However, there are minor differences between the two FELIX systems, even if the hardware for both experiments is the same. In the case of the DUNE experiment, data are streamed into physical memory and kept there for processing, whereas in the ATLAS experiment data are published by the FELIX system through a switched network for access by the clients (Dataflow nodes). In practical terms, the readout in DUNE is performed by a single physical unit containing the FELIX system, whereas in ATLAS the FELIX system is separated from the readout nodes (Data Handlers).

Moreover, from a readout perspective, it is also worth noting that the DUNE-FD module adopts a continuous readout strategy in which the front-end electronics send data to the readout system at a fixed rate for the TPC detector, regardless of the data content. On the other hand, the ATLAS experiment takes only the data that has successfully passed the first level of filtering. In addition, collider experiments always have front-end data organized into fragments belonging to a certain event identifier, usually corresponding to the proton bunches crossing inside the detector. This is quite different from the DUNE experiment where there is no indication for the readout system on how to group front-end data into fragments corresponding to the same physical event.

Finally, both the DUNE and ATLAS experiments use high-performance storage buffers in the DAQ architecture. In the case of the DUNE detector, the objective is to provide a local storage system within the readout nodes for a high-throughput application that needs to be active once a rare trigger signal is received. In the case of the ATLAS detector, a large distributed storage buffer is planned to decouple the live data-taking from the event filtering. Despite the fact the technologies planned for both storage systems may be the same, their use-case in the data acquisition system is different. Therefore, extensive research is needed to investigate storage technologies and dataflow techniques for the development of the data acquisition system in the context of both the ATLAS and DUNE experiments. These topics will be discussed in detail in the following chapters.

7

High-performance storage buffer for supernova events

7.1 Introduction

Emerging high-performance storage technologies are being used to fill the gap between memory and traditional storage. An example of these technologies is the 3D XPoint which provides high-performance storage media in the form of of innovative persistent memory devices and storage devices (e.g. Micron® X100 SSDs). The objective of this chapter is to showcase the performance characterization of different high-performance storage technologies as a potential application for the supernova storage buffer (local storage) of the DUNE data acquisition system. Note that, unless otherwise stated, DUNE is referred within the context of the Far Detector horizontal drift module. The material used for this chapter is mostly taken from [21] and [22].

7.2 Physics use-case: a storage buffer for supernova events

One of the physics objectives of the DUNE experiment is to study neutrinos originating from galactic sources. It has been shown in [23] that liquid argon TPC detectors are capable of detecting neutrino bursts from core-collapse supernovae, also known as supernova burst events (SNB). The neutrinos from cosmic sources are expected to have an average energy less than

30 MeV. From the experimental perspective, the detection principle is based on the elastic scattering on electrons from the different flavours of neutrinos ($\nu_x + e^- \rightarrow \nu_x + e^-$ where $x = e, \mu, \tau$). It is also possible to detect events from de-excitation of γs in the absorption process of ν_e on argon nuclei ($\nu_e + {}^{40}\mathrm{Ar} \rightarrow e^- + {}^{40}\mathrm{K}^*$; $\bar{\nu}_e + {}^{40}\mathrm{Ar} \rightarrow e^+ + {}^{40}\mathrm{Cl}^*$). Note that this last process is sensitive to only the ν_e neutrino species.

Figure 7.1 shows the cross section of different interactions as a function of the neutrino energy for several processes relevant to supernova events. Note that the ν-e cross section, main signature of neutrinos from supernovae, is several orders of magnitude lower than the ν-Ar cross section. Therefore, detecting events from cosmic neutrino sources is quite challenging and large detector volumes are needed in order to produce enough interactions to record a statistically significant sample of events. This is because the expected experimental signature of a supernova burst event is characterized by multiple low energy deposits distributed across the whole active detector module. As an example, a 3 kt experiment such as ICARUS will be able to detect approximately 250 events from a supernova burst event originating from a distance of 10 kpc [23].

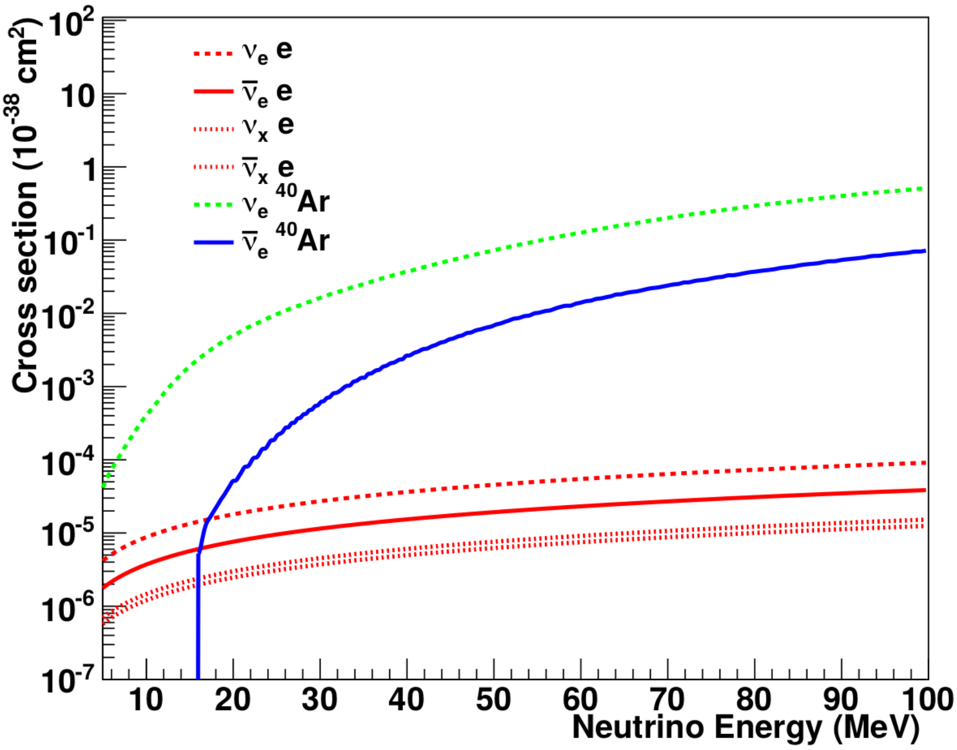


Figure 7.1: Relevant cross sections for supernova burst events for several interactions in argon. ν-e cross sections represent elastic scattering interactions, whereas ν-Ar represent absorption interactions. Figure taken from [3].

From the data acquisition perspective, a longer trigger latency is needed to identify an SNB event because neutrino interactions are accumulated over a certain amount of time across the detector. It is expected that to fully reconstruct an SNB event, the entire detector must be read out for up to 100 seconds [9].

The trigger selection in DUNE is typically based on clustering hits from the TPC collection wires of the DUNE-FD Horizontal Drift module. To claim a neutrino interaction from a supernova, a hit finder algorithm is applied to select only the hits that have deposited a charge above a certain threshold. Such hits are then grouped in clusters. The supernova trigger selects only clusters that satisfy the requirements of being on consecutive channels (wires) and close in time [24]. The efficiency of this triggering technique was studied in different configurations of signal-noise ratio to satisfy the DAQ requirements and have the least amount of fake SNB events as possible. The current best estimate for a fake SNB event is one per month. Further details on the triggering of SNB events can be found in [25].

7.3 Supernova storage buffer: objective and requirements

As discussed previously, the data acquisition of the DUNE-FD Horizontal Drift module will consist of a large-scale distributed system designed to handle a total of 1.5 TB/s of incoming data from the readout system. The DUNE baseline design for the readout system consists of approximately 80 server nodes with installed 150 custom PCIe cards (FELIX I/O devices), each receiving data over ten 10 Gb/s optical links and streaming data into the host memory at a rate of 1 GB/s per link. Data are buffered in physical memory (DRAM) for about 10 s to allow a sufficient readout window ranging in the past for SNB triggers. This is done in order to identify also the start of the neutrino flux which may not have triggered. Data are kept in memory until either a trigger signal is received and selected data are sent out to the event builders over the network or data have aged beyond 10 seconds and, thus, they are discarded.

Upon the arrival of the supernova trigger, a high throughput data path to storage is activated in the readout units (RUs). This allows to save in the order of 100 s of continuous data, which then will be forwarded at a slower pace to the event builders. Persistent data buffering is required because of the value of those data (rare physics signals) and because their transfer over the network will take several hours. A subsequent power cut, a server reboot or an application crash will not cause any data loss if the events are locally stored in the readout nodes.

Finally, the trigger is configured with thresholds such that statistical fluctuations will fire the

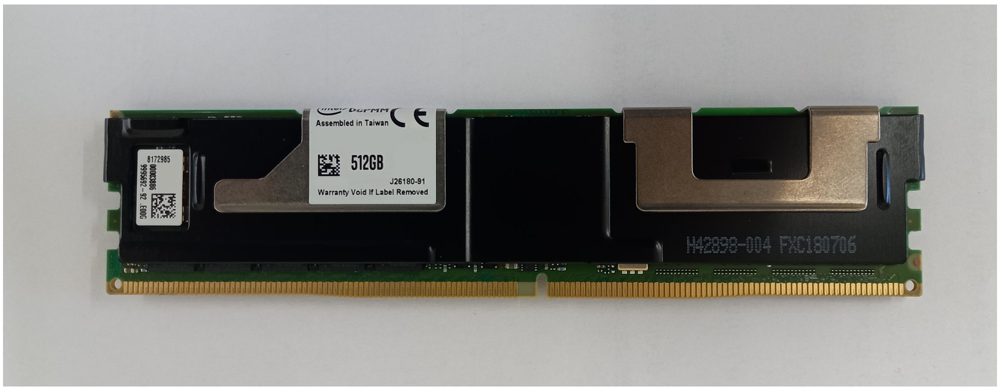


Figure 7.2: Image of a DCPMM device.

supernova burst trigger in the order of once per month. A recording time of 100 s of data correspond to 1 TB for each CPU socket of the RU. The baseline requirement is to be able to store up to two such data complements. Therefore each CPU socket of the RU will need a total of 2 TB of usable local storage space for the supernova buffer.

7.4 Persistent memory modules

Given the importance of the data being recorded, DRAM technology is not a viable solution for the DUNE local storage buffer as it cannot provide storage persistence. In addition, the total buffer size needed for each server would make the system too costly with only DRAM modules. One possible way to achieve the target objective for the system is to use fast storage media: an example of such devices are the Intel® Optane™ Data Center Persistent Memory Modules (DCPMMs or PMEMs) which leverage the 3D XPoint technology. Figure 7.2 shows an image of a DCPMM device.

7.4.1 3D XPoint memory technology

3D XPoint™ is a memory (NVM) technology that has been jointly developed by both Intel® and Micron® Technology Inc.®. This is a type of non-volatile technology that offers approximately 10 times higher bandwidth compared to traditional NAND-based storage media [26]. Intel® Optane™ devices are based on the 3D XPoint™ memory technology. They have been built both as memory module devices and as solid-state drives (SSDs).

Non-volatile memory (NVM) is an emerging and innovative technology which uses the memory bus, like commonly available DRAM modules. DCPMMs are non-volatile memory devices that use the 3D XPoint™ technology. They offer memory-like performance at a lower cost per gigabyte compared to DRAM and provide higher disk writes per day DWPD compared to other storage media (further details are provided in chapter 8). Therefore, DCPMMs are good candidates to fill the performance gap between memory and storage devices. In addition, contrary to storage media where data access is usually done with a 4 KiB block size, DCPMMs fetch the data with 4 cache lines of 64 bytes. This results in a 256 bytes load/store instruction that provides a lower latency, similar to memory devices.

7.4.2 Operation modes of DCPMMs

From an operational point of view DCPMMs can be configured into three modes:

- *Memory mode*: in this mode the DCPMMs act as a large memory pool alongside the physical memory modules. DCPMMs are seen by the operating system as a large volatile memory space

- *App Direct mode*: in this mode the DCPMMs provide in-memory persistence. Therefore, they act as storage devices rather than memory devices. The memory controller maps the DCPMMs to the physical memory address space of the machine so that the software layer can directly access the devices

- *Mixed mode*: in this mode it is possible to use a percentage of the DCPMMs capacity as both memory and storage devices (mixed Memory and App Direct modes)

In addition, in the App Direct mode the DCPMMs can be configured into two ways:

- *Interleaved region*: all the DCPMMs relative to a CPU socket are seen as a single block device as if the modules are used in a RAID 0 configuration

- *Non-interleaved region*: each DCPMM is seen as a separate block device and, therefore, each module can be accessed independently

Finally, depending on the configuration, it is possible to mount the DCPMMs with a *direct access* file system (DAX). This provides byte-addressable access to the storage without the need to perform an extra copy on the page cache and, thus, it yields higher read and write bandwidths.

Table 7.1: Overview of the test machine used for the DCPMM evaluation.

CPU	Intel® Xeon® Platinum 8280L 2.70 GHz (Cascade Lake), dual socket, 28 cores L1 cache 32K L2 cache 1024K L3 cache 3942K
DRAM	DDR4 DRAM 16 GB, 2666 MT/s, 12 slots Product number: Kingston KSM26RS4
DCPMM	DDR-T 512 GB, 2666 MT/s, 12 slots Product number: Intel®NMA1XBD512GQS
OS	CentOS 7, Linux Kernel 4.15.0[1]
SW	ipmctl v.01.00, PMDK v.1.9

7.4.3 System description

Testing environment

Table 7.1 summarizes the specification of the machine node used for the evaluation. The test machine is a dual CPU socket system with 28 physical cores on each processor and one memory controller per socket. Each memory controller has 6 memory channels composed by both a DDR4 DRAM device (16 GB) and a DCPMM (512 GB). The total DRAM size of the machine is 192 GB whereas the total DCPMM size is 6 TB. Finally, the node has a CentOS 7 operating system with kernel version 4.15.

Testing strategy

The raw performance of the persistent memory devices was obtained by executing a synthetic benchmark evaluation with the DCPMMs used as a storage device in App Direct mode and mounted with a DAX-enabled ext4 file system. Instead of relying on low-level benchmarking tools that are highly customized for storage devices, the DCPMMs were tested with a C++ application. This was developed in order to control the tuning parameters and to perform the testing in a realistic environment which resembles as much as possible a production-ready application. In the evaluation, the throughput was measured as a function of both the block size and the number of threads. The resulting performance was obtained in terms of request rate and in terms of total throughput. The objectives of such testing was to understand the limitations of the underlying storage hardware and to identify the optimal tuning parameters that maximize the sequential write throughput. Sequential write is, in fact, the ideal access pattern for the implementation of the DUNE local storage buffer because a SNB event can be translated into a large continuous file (at least 150 TB) that is extracted from the memory buffers of the readout node and written to adjacent locations on the storage media.

The results obtained were reproducible across several runs. However, a complete analysis of the uncertainty on the individual measurements was not computed as it was not relevant for the objectives of this research because the goal was to assess the suitability of the DCPMMs as a possible technology candidate for the DUNE storage buffer. Therefore, the interest was focused on the scaling of the system and the observed minimum/maximum values.

The benchmarks executed on the system refer to the 6 DCPMMs (App Direct mode with interleaved configuration) connected to the same CPU socket unless otherwise stated. This is to ensure that the results are compatible with the requirement of sustaining the data rate of a single FELIX card per CPU socket.

7.4.4 Benchmarks of DCPMMs

Synthetic benchmarks of the DCPMMs are a good tool to test their feasibility as possible technology for the DUNE supernova buffer. In fact, they give a quick indication whether the devices under investigation are capable of sustaining the required data rates.

Figure 7.3 represents the request rate as a function of the I/O block size for different writing threads. This was obtained by measuring the number of I/O operations per second for a given workload in terms of block size and number of threads. Increasing the block size results in a smaller request rate due to the increased latency to fetch and store the requested block. The maximum rate sustained by a single writing thread and I/O block size of 4 KiB is approximately 250k operations per second per thread. This showcases the small time needed to request data with DCPMMs. Interestingly, the figure highlights that in order to achieve the highest rate from the DCPMMs it is necesary to tune both the number of threads as well as the block size.

Figure 7.4 shows the throughput as a function of the number of threads in the case of both the reading and writing access pattern. The block size used for the benchmark is 256 bytes because it represents the lowest access granularity for DCPMM devices. These results were obtained by executing a writing (or reading) thread on all the DCPMMs (12 modules in total) using an interleaved region and measuring the time taken to write (or read) the selected data block size (synthetic testing). Note that CPU affinity[2] was set up on the host in order to restrict the executing threads to the physical cores of the same NUMA node of the DCPMMs.

[2] The CPU affinity is the binding of one or multiple processes to a specific CPU core.

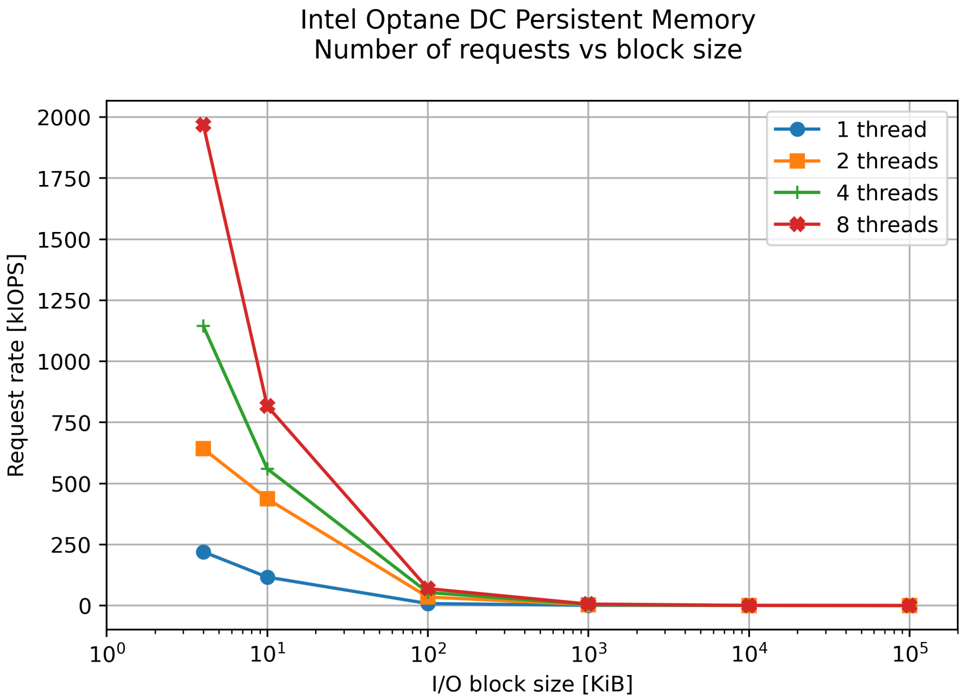


Figure 7.3: Request rate as a function of the I/O block size for different writing threads. Increasing the block size results in a smaller request rate due to the increased latency to fetch and store the requested block.

This was done to avoid any cross-NUMA access that leads to an increased latency and, therefore, lower performance. The configuration adopted for this synthetic benchmark resembles the workload expected for the DUNE local storage buffer. In fact, the ideal application for the supernova storage would need to sustain the data output from ten optical links. Therefore, writing to ten separate threads represents the ideal configuration. In addition, in the synthetic benchmark the writing thread was executed by memory mapping the block of data and then using the MOVNTI [27] non-temporal SSE instruction [28] provided by the CPU processor. In this way, the operation has no overhead from the file system because it invalidates the cache line and therefore results in a pure device operation. This feature is possible because it is allowed by DAX-enabled file systems. In this way, it is possible to achieve higher bandwidths. Figure 7.5 shows a schematic representation of the difference between running an application on a traditional file system compared to a DAX enabled file system. As it can be noted, on a DAX file system there is no page caching within the kernel-space as compared to the behaviour of a traditional file system.

From figure 7.4, the maximum write throughput obtained from the benchmarking application is approximately 8.9 GiB/s. The maximum write throughput is obtained with 12 threads, suggest-

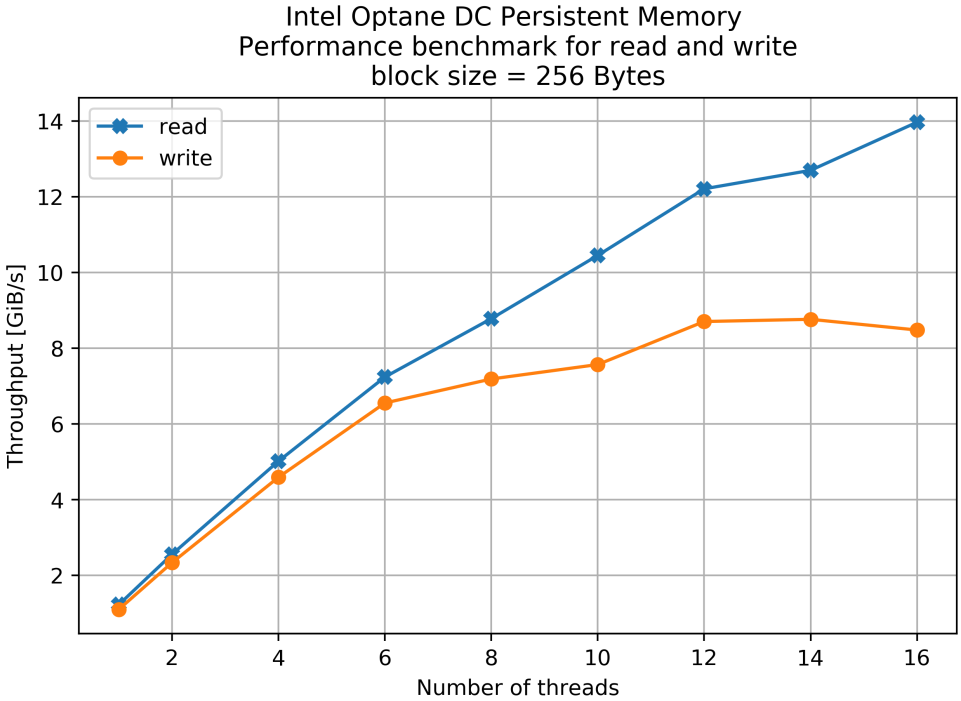


Figure 7.4: Throughput as a function of number of threads for a block size of 256 bytes in the case of both sequential reading and sequential writing.

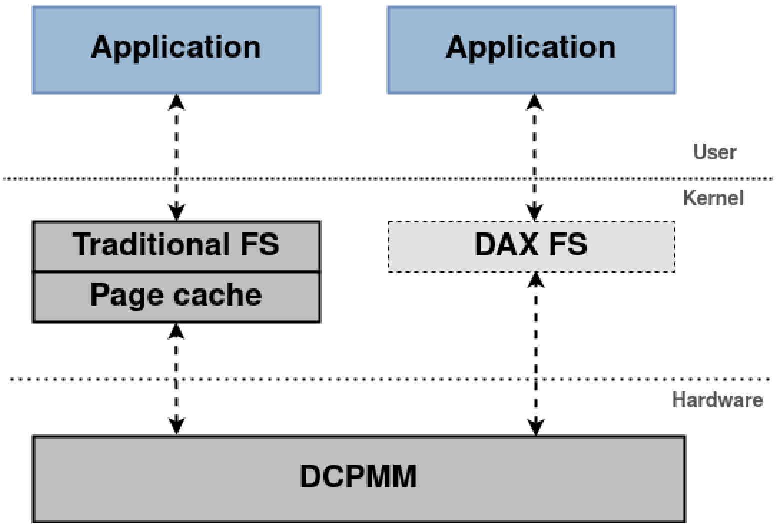


Figure 7.5: Schematic view of an application running on DCPMMs with a traditional file system and with a DAX-enabled file system.

ing that the DCPMMs have reached their maximum rate limit. More threads will not increase the throughput as the devices have become latency-bound (or limited by their rate). In the case of the reading access pattern, the throughput is higher and it has not reached a plateau even when testing with 16 threads. This is because DCPMMs also behave as memory devices and, thus, have lower reading latencies compared to the writing operation. Therefore, it is possible

Table 7.2: Bandwidth for a 100% sequential read and 100% sequential write workload for both a NAND-based SSD and DCPMMs.

	Bandwidth **NAND-SSD [GiB/s]**	**Bandwidth** **DCPMMs [GiB/s]**
Sequential read	3.2	40
Sequential write	1.9	8.9

to achieve higher bandwidths. As shown in [29], in the best case configuration, the maximum achieved throughput for a read workload is approximately 40 GiB/s.

Finally, a comparison between the maximum bandwidths provided by both DCPMMs (12 modules) and a NAND-based SSD is illustrated in table 7.2. The NAND-based SSD is a PCIe Gen 3 Intel® SSD DC P4510 (2 TB) which was benchmarked with a similar approach to the DCPMMs using a C++ application performing a reading or a writing operation. The results obtained for the NAND SSD are also confirmed on the device's technical data-sheet [30]. Note that the sequential read bandwidth for DCPMMs is taken from [29]. This shows that in order to achieve a write bandwidth similar to the one provided by the DCPMMs it would be necessary to use almost 5 NAND-based SSDs. It is also worth mentioning that the new generation of SSD devices utilizing the PCIe Gen 4 interface [31] are capable of sustaining higher bandwidths, therefore fewer devices are needed to match the performance of DCPMMs. However, the potential use of such technology, i.e. using multiple SSDs together, has to be carefully evaluated with the availability of PCIe lanes on the host server. In fact, employing four or five 4-lane PCIe based devices may have a considerable impact on the total number of available PCIe lanes on the host server.

7.4.5 Emulator for the ProtoDUNE data acquisition

The ProtoDUNE-SP experiment is used as a testing platform in view of the future DUNE Far Detector. As mentioned previously, the readout system of ProtoDUNE uses a FELIX I/O card that generates approximately 10 GB/s, per CPU socket, from ten optical links and, upon receiving a trigger signal, data need to be stored for 100 seconds. In addition, data from the FELIX system are kept in volatile memory until a trigger signal is issued. This means that the storage buffer for the DUNE data acquisition system needs a technology candidate capable of sustaining a writing rate of approximately 10 GiB/s to avoid over-sizing the physical memory of the host nodes. Figure 7.6 shows how the size of the (volatile) memory buffer would need to increase as a function of time depending on different writing bandwidths (output rate) of a generic storage solution. For example, suppose the writing bandwidth of the selected storage technology is only 5 GiB/s. In that case, the total physical memory of the system needs to be increased to 500 GiB to keep up with the input rate and for the total time of 100 s. Therefore, it is crucial to find a suitable storage solution capable of sustaining the target rate to minimize the extra volatile memory needed to keep the data.

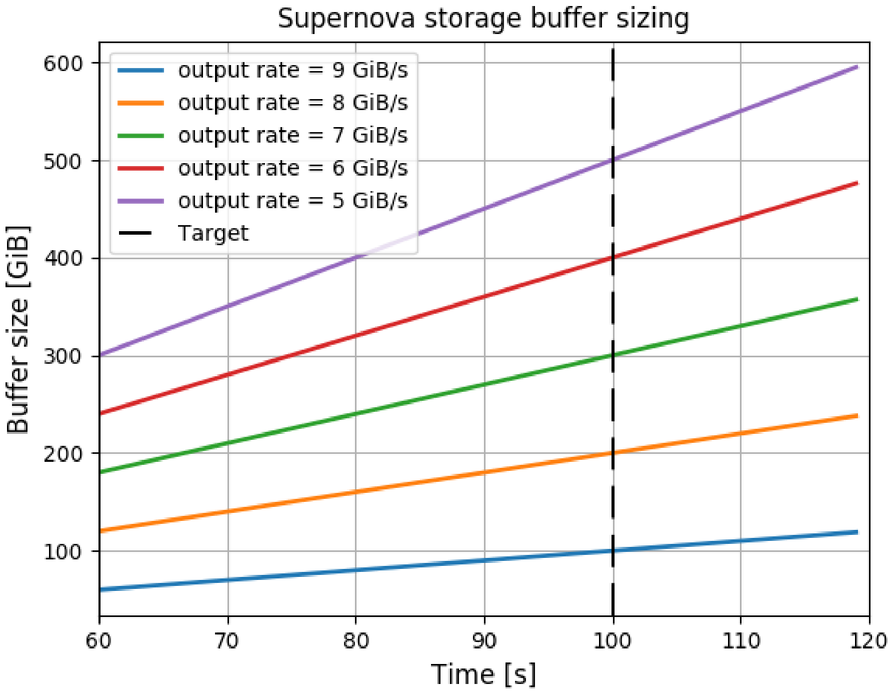


Figure 7.6: Amount of DRAM as a function of the Data Recording time for different output rates. Decreasing the output rate requires an increased memory size. A vertical line representing the target of 100 s is also included.

Based on the synthetic benchmarks, the DCPMMs represent a good candidate for the DUNE supernova storage system. To test the feasibility of the DCPMMs for such a system, an emulator of the readout workload was developed and used for testing. From an implementation perspective, the testing setup consists in multiple sequential writing threads using a block size of 5568 bytes. This is the superchunk block size used by the FELIX card after grouping 12 WIB frames and it represents the smallest access size used for transferring and storing the data. In addition, only six DCPMMs were configured in the interleaved mode to check whether the devices under investigation can sustain the emulated workload. Finally, the test application has been written using the Persistent Memory Development Kit (PMDK) [32] which is a collection of libraries and tools that ease the development of high-performance applications that use persistent memory devices.

Figure 7.7 illustrates the performance obtained with the emulated workload application. The figure shows the average throughput for an increasing number of threads and a block size of 5568 bytes. The result obtained in the test application is satisfactory up to four threads because the average throughput per thread can sustain the target bandwidth (per link) of 1 GB/s. However, as the threads number increases, the application cannot keep up with the incoming rate because the total throughput decreases to 870 MiB/s which is less than the target value. As a consequence, the software stack of the workload emulator was optimized in order to fully exploit the performance provided by DCPMMs. It was noticed that the *libpmemblk* library of the PMDK tool was adding extra overheads to the application because of features such as block-level atomicity in case of errors or power failures which are not needed for the DUNE supernova storage buffer. Similar to the benchmarking evaluation of the DCPMMs, a lower-level application was developed by creating memory-mapped files and then persisting them using the MOVNTI CPU instruction. In this way, there is no extra overhead from the file system and the performance obtained is higher. By optimizing the software stack of the storage application, the throughput obtained in the emulator application (figure 7.7) increased by approximately 17% when the PMDK library was not used. With such an optimization the DCPMMs can be considered suitable devices for the implementation of DUNE local storage buffer.

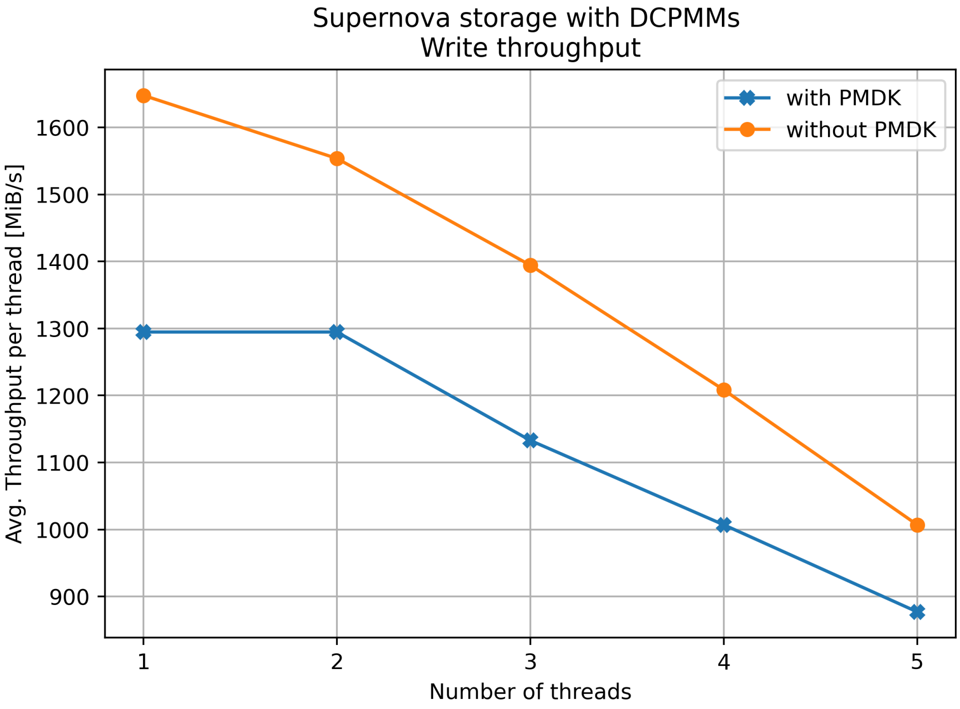


Figure 7.7: Average write throughput per thread as a function of the number of threads for the ProtoDUNE test application with six DCPMMs. Results obtained with and without the PMDK software.

7.4.6 Integration in the ProtoDUNE data acquisition

A second application was also developed and integrated with the data acquisition system of the ProtoDUNE experiment. The objective is to have an application resembling as much as possible the DUNE workload: a traffic pattern is generated by memory copying data into volatile memory (emulating the behavior of the FELIX device) and then, upon receiving a command, persisting the data into the storage media (emulation of the supernova trigger). All the memory DCPMMs available on the system on both sockets have been deployed in the App Direct mode. Figure 7.8 illustrates the throughput obtained as a function of the number of executing threads for both the interleaved and the non-interleaved DCPMM configuration. The system saturates the available bandwidth with a throughput of more than 7.5 GiB/s starting from 4 threads. Therefore, with the current DCPMMs available today it is possible to sustain, per CPU socket, only 75% of the target throughput required for the DUNE supernova storage buffer.

The non-interleaved configuration of the DCPMMs was also tested because it represents a good match for the DUNE supernova workload. In fact, the ten writing threads can be considered independent as they refer to different optical links and, therefore, can be configured to write to 10 separate block devices. However, as shown in figure 7.8 the throughput obtained in this operational mode is much lower compared to the interleaved configuration. For example, in the case of 5 writing threads the throughput in the non-interleaved configuration is approximately 60% lower than in the corresponding interleaved DCPMM region. In addition, in the case of the non-interleaved configuration the maximum throughput that the DCPMMs can sustain has not been achieved with 10 threads. This behavior confirms that the interleaved configuration is the most suitable for high-performance applications[3] and it suggests that DCPMMs in this configuration have internal mechanisms to optimally balance the I/O operations to achieve the best performance.

7.4.7 Integration with the DUNE-DAQ software

Another testing platform is the integration of the high-performance storage buffer with DCPMMs within the DUNE-DAQ software. As introduced in chapter 5, the Readout system of the DUNE-DAQ is designed with a buffering and storage domain responsible for storing the data frames. The latency buffer of the Readout system is designed to temporarily store the data in physical memory. However, a local data store is also included for recording the full stream of

[3] This is also confirmed in the technical sheet document of DCPMMs

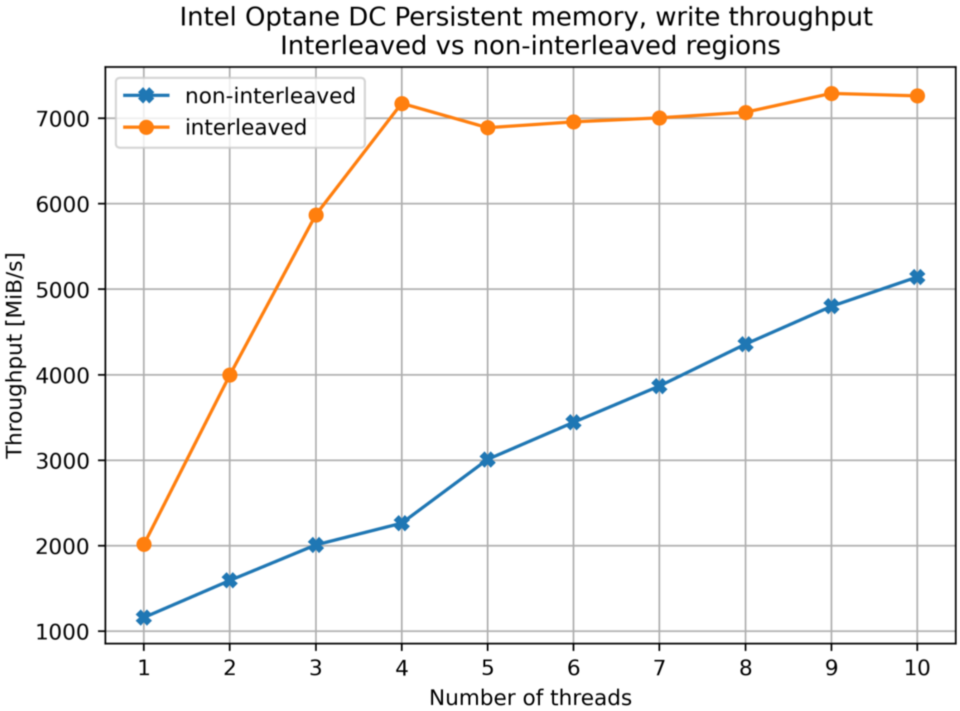


Figure 7.8: Write throughput as a function of number of threads for both interleaved and non-interleaved DCPMM configuration for a DUNE-like application. Maximum throughput is obtained at 7 GiB/s starting from 4 writing threads.

raw data frames in persistent storage media.

The Data Recording feature of the DUNE-DAQ software provides a high-performance memory-aligned file writer application. This is a generic implementation that can support different media for the local data store such as a RAID of NMVe SSDs or Non-Volatile Memory Modules (DCPMMs). In particular, the DCPMMs have been been mounted with the interleaved operation mode and configured as the target data store of the Data Recording application within the DUNE-DAQ readout package. Therefore, no specific code for DCPMMs was used for this integration testing. In fact, the DCPMMs were used only as storage devices without deploying custom applications that use specific instructions for DCPMMs.

The integration consists of three testing phases. The first phase includes a *fake readout* application that emulates the behavior of a FELIX I/O card in the same way that has been done when testing the DCPMMs in section 7.4.5. In the second phase, a FELIX I/O card was mounted on the same server hosting the DCPMMs and it was configured in *emulator mode*. In this configuration, the FELIX I/O card is able to generate packets with a total aggregated throughput of 8.8 GB/s for 10 data links. Finally, the last phase of the integration testing consists of using a

FELIX I/O card that is connected to the WIBs of the ProtoDUNE-SP detector. This makes it possible to test the system's behavior in a realistic scenario. During the three integration steps all the relevant DUNE-DAQ applications (e.g. dataflow, DQM, data selection, run control, etc.) were executed concurrently to investigate if the Data Recording application is capable of running concurrently with the other DUNE-DAQ applications in a realistic scenario expected for the DUNE detector.

The objective of the testing is to investigate whether the DCPMMs can be a viable solution for the local storage buffer when using the latest version of the DUNE-DAQ software[4] and by running the whole data acquisition chain: raw data generation, data recording, data request. For this integration, a test is considered successful if there is no back-pressure in the system[5] or no rate drops are observed because of potential failures during the data taking. In addition, the data recording has to be executed for at least 100 seconds for it to be successful.

The testing with the emulated FELIX card showed the same throughput results obtained in section 7.4.5 and it confirmed that the DCPMMs are suitable devices for the DUNE local storage buffer. The integration testing with the FELIX I/O card (either in emulated mode or connected to the WIBs) was also successful: no rate drops or errors were observed during the testing. This means that DCPMMs are capable of fully sustaining the target throughput of 8.8 GB/s for at least 100 seconds. This is in fact close to the maximum bandwidth provided by DCPMMs. It is worth mentioning that a lot of effort was invested into separating the running threads of the several DAQ applications to the available cores of the host server. In particular the data recording application and its threads were carefully set to isolated physical cores where no other thread is running and they were set on the same NUMA node of the DCPMMs. In addition, the data recording application (file writer) used in the DUNE-DAQ integration tests uses memory-aligned allocations and performs I/O operations asynchronously bypassing the operating system page cache (O_DIRECT kernel flag). Thanks to the several optimizations in place, the file writer of the DUNE-DAQ software represents a high-performance application that fulfills the requirement of the Readout system to persist the full raw data stream for at least 100 seconds (supernova storage buffer).

[4] DUNE-DAQ version 2.9.0 was used at the time of writing.

[5] Condition that occurs when a sub-system/buffer gets saturated and the incoming load is transferred to another sub-system or other buffers become saturated.

Table 7.3: Overview of the test machine used for the evaluation.

CPU	AMD® Epyc® 7302 Dual socket, 16 cores
DRAM	DDR4 DRAM 16 GB, 3200 MT/s, 32 slots
OS	Centos 7, Linux Kernel 5.10
Storage	Micron® X100 SSD (750 GB)
SW	FIO v3, AIO library
NOTE	All CPU cores have been set to *performance* mode

7.5 A novel high-performance NVMe drive

The 3D XPoint™ technology has also been developed for PCIe-based NVMe SSDs. The Micron® X100 is an example of such a device and it was selected for the evaluation because it provides a nominal bandwidth closer to the one needed for the DUNE supernova storage buffer. As for the DCPMMs, a first synthetic evaluation of the Micron® X100 devices was performed and, subsequently, a testing application was integrated into the prototype software of the DUNE DAQ.

7.5.1 Description of the setup and the tools used

The test machine used for the synthetic benchmarks is characterized by a dual-socket CPU processor (16 cores) and 32 DDR4 DRAM DIMMs, each of 16 GB. The storage device used for testing is a Micron® X100 SSD drive based on the PCIe Gen. 3 (16 lanes).

Proper server configuration setup was needed in order to achieve the highest performance from the storage devices. This was achieved by setting a proper thermal management system (e.g. setting maximum fan speed from the BIOS) and setting all the CPU cores to *performance* mode. Table 7.3 summarizes the specification of the machine node used for evaluation.

For the synthetic benchmarks, a first evaluation was done using the *flexible-IO* (FIO) tool[33]. This is a standard tool used to benchmark storage devices as it allows users to tune the tests to achieve an emulated workload as close as possible to the final application. In addition, *fio* also provides access to low-level parameters (e.g. access pattern, size of the storage queues, inflight operations, etc.) that make it easier to tune and achieve the highest performance from the storage hardware. Note that this tool is generally used to benchmark block devices and, therefore, it cannot be readily adopted for the DCPMM devices.

Another tool that was used in the evaluation is the *libaio* library [34]. The main motivation for using this tool, in addition to *fio*, is to have a slightly more realistic, higher-level application that resembles the workload expected in the DUNE data acquisition system. The main reason to use libaio compared to other tools is motivated by the need to have a high-performance storage library capable of efficiently writing data to NVMe devices. The libaio library performs asynchronous file operations by relying on the native kernel AIO interface and thus it appears to be closer to the hardware device rather than performing operations from user-space. The asynchronous nature is achieved by using the O_DIRECT kernel flag which allows to bypass the operating system page cache. In this way, the write operation inserts the data directly into the storage device's queue which can then process the input.

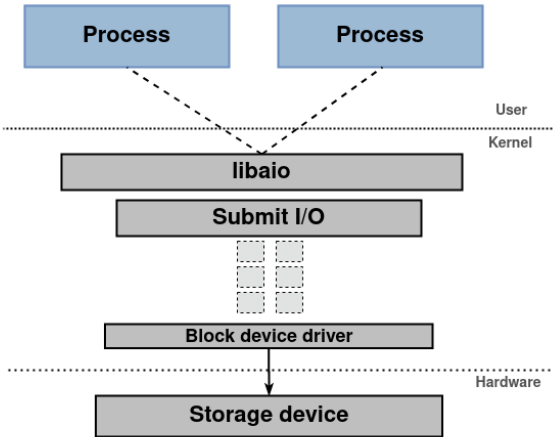


Figure 7.9: Schematic representation of the libaio workflow highlighting the separation between user-space, kernel-space and hardware domains.

Figure 7.9 shows a schematic representation of the typical workflow when using libaio, highlighting in particular the separation between the tasks in user-space, in kernel-space and in hardware. Another high-performance I/O library is POSIX AIO: this library uses blocking I/O with multiple threads in user-space to create an asynchronous pattern. The main disadvantage is that POSIX AIO may lead to a considerable large memory footprint [35]. In addition, POSIX AIO cannot support multiple AIO requests on a single file descriptor and, therefore, is not a good tool for developing the application for the DUNE supernova storage buffer.

7.5.2 Synthetic benchmarks

In this section, the performance evaluation of the Micron® X100 SSD is illustrated in terms of its achieved sequential write throughput in view of its application for the DUNE local storage buffer. The novelty of this research lies in evaluation of a newly-introduced device which was tested from an application perspective without relying only on low-level benchmarking tools.

As in the case of the DCPMMs, the synthetic evaluation of the Micron® X100 device was performed by carefully setting the CPU affinity of the executing thread to the corresponding NUMA node on which the Micron® X100 SSD was installed.

Figure 7.10 illustrates the sequential write throughput of the Micron® X100 SSD as a function of the block size for a single thread. The scan was performed starting from a block size of 4 KiB up to a block size of 32 MiB. The maximum performance that the drive is capable of sustaining is more than 8.5 GiB/s. The block size at which the peak performance is reached was achieved starting from a block size of 8 MiB. This suggests that, in case of a single thread writing to the Micron® X100 SSD, it is necessary to use large block sizes to achieve the highest throughput.

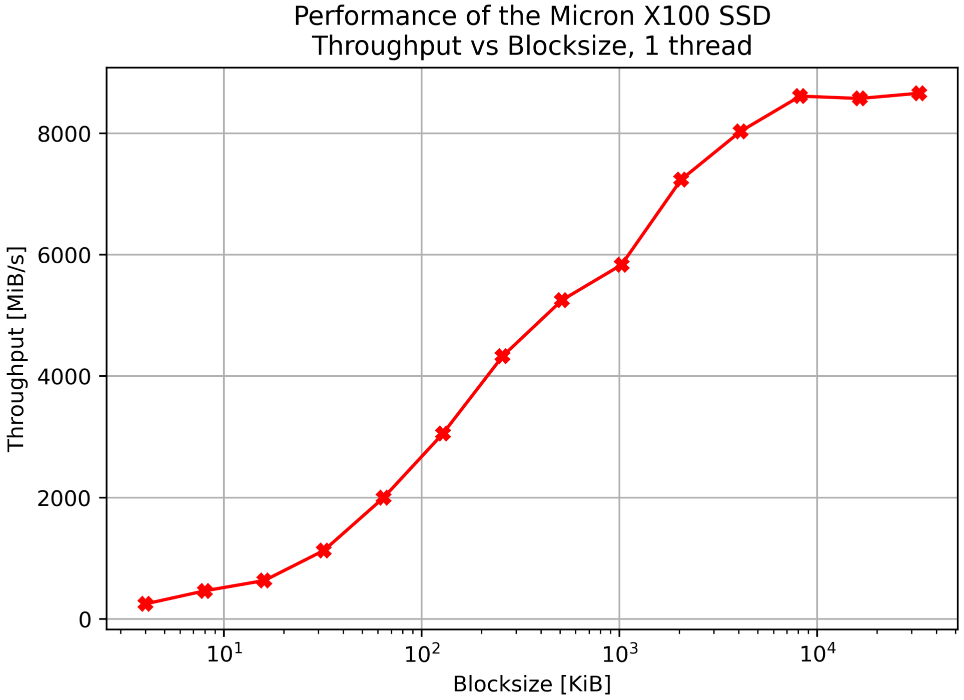


Figure 7.10: Write throughput as a function of the block size ranging from 4 KiB up to 32 MiB. The system reaches the maximum bandwidth with a block size of around 8 MiB. Logarithmic scale on x-axis.

The throughput as a function of the number of threads is shown in figure 7.11. This measurement was done by issuing up to 10 threads, which is the expected number of links that the readout system needs to sustain. A scan in block size was also executed, ranging from the KiB regime up to the MiB regime. The results show that increasing the number of writing threads makes it possible to achieve the maximum throughput with smaller block sizes. For example, with a block size of 32 KiB, the throughput achieved with ten threads is four times higher than that achieved with only one thread. Thus, this confirms that to fully exploit the drive's performance with block sizes smaller than 4 MiB a multi-threaded approach is necessary. It is also worth noting that the measurement was performed using direct I/O, which is a feature of the Linux file system, to bypass the operating system cache. Therefore, in this way, it is possible to write directly from the application to the storage device and measure its raw performance. In addition, when performing the test with a block size of 4 MiB the system CPU utilization (per thread) was approximately 15% for the whole testing time. This originates from the *libaio* library which relies on kernel calls to perform the asynchronous operations and it indicates that the throughput of the application is not bound by the CPU load but by the physical performance of the storage drive.

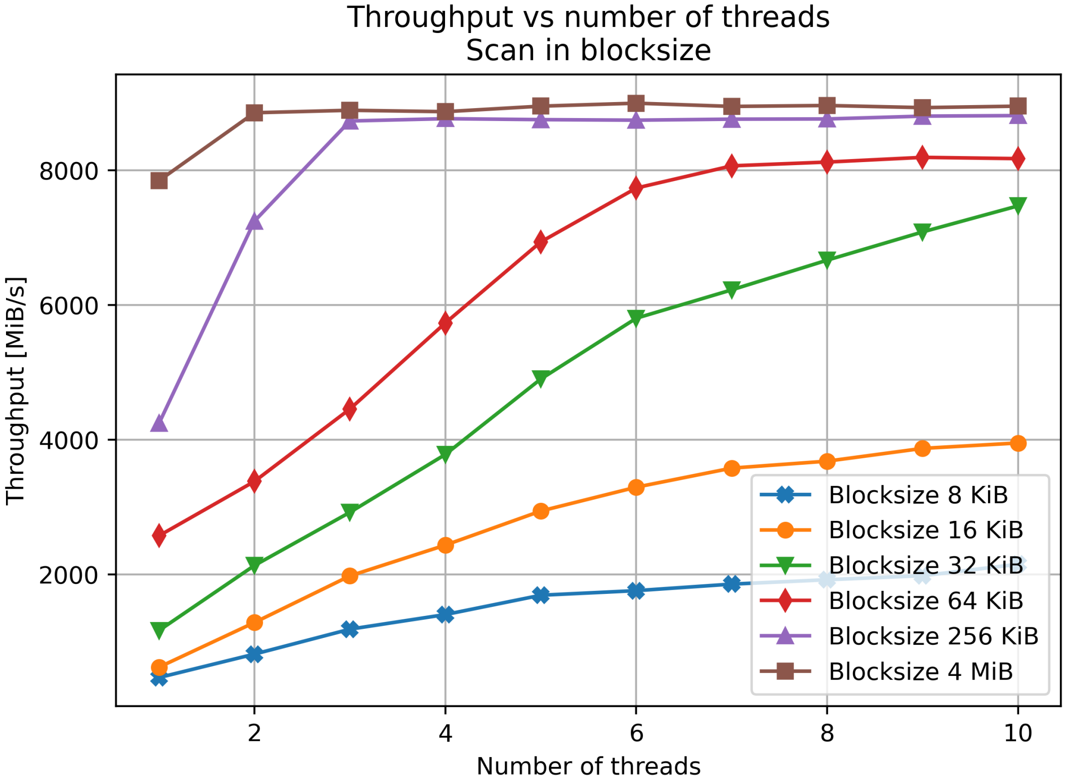


Figure 7.11: Write throughput as a function of the number of threads ranging from 1 to 10. A scan in block size has also been performed from 8 KiB up to 4 MiB.

The synthetic evaluation has shown that the throughput achieved with the Micron® X100 device is less than the nominal throughput of approximately 8.8 GB/s needed to fully build the DUNE local storage buffer. Nonetheless, the Micron® X100 still represents an interesting device for the supernova storage system of the DUNE experiment.

7.5.3 Integration with a prototype of the DUNE-DAQ

Although the results from the benchmarking testing of the Micron® X100 device show a good match with the specification of the hardware drives, an evaluation with a more realistic workload was performed to assess the suitability of the Micron® X100 drives for the DUNE experiment. This was done using a test application (*MiniDAQ* version 1.2.0) that contains the most relevant features needed to emulate the final data acquisition system. Figure 7.12 illustrates the main components of the *MiniDAQ* application: Readout (grey), Dataflow (blue) and Trigger (green). The Readout Emulator emulates the data input from half of a FELIX readout board (only 5 links). The Data Link Handlers manage the temporary buffering of raw data for each link. The Trigger Decision Emulator acts as the data selection system. Trigger decisions are translated into data requests by the Request Generator and the Data Link Handlers respond to any data requests by extracting the raw data, formatting them and forwarding them to the Fragment Receiver. The Fragment Receiver aggregates data corresponding to the same trigger decision and forwards data to the Data Writer, which implements the interface to the permanent storage. Each Data Link Handler receives a stream of 5568 bytes at rate of 166 kHz which corresponds to the data size and rate expected in output from the FELIX board. Two instances of the *MiniDAQ* application were used in parallel to emulate the traffic of one DUNE readout board. Note that the *MiniDAQ* application used for the Micron® X100 integration testing has the same functionality to the DUNE-DAQ software used for the integration of the DCPMMs. The main difference between MiniDAQ and the DUNE-DAQ software is that in the former data are extracted by the Datatflow system, whereas in the latter data are directly streamed from the readout's memories to the storage buffer.

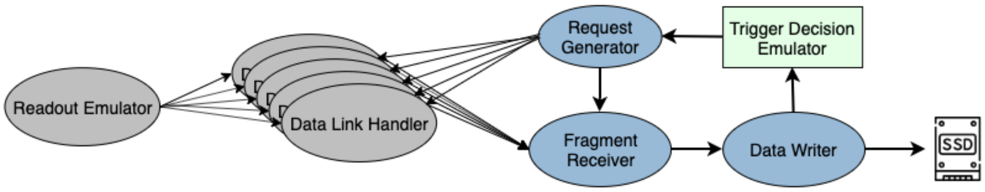


Figure 7.12: Schematic representation of the main elements that constitute the MiniDAQ test application.

Algorithm 1: Data Writer mechanism.

```
allocate mem-aligned buffer;
start data writer thread;
set CPU affinity;
while trigger_flag do
    receive requested_data;
    for fragment in requested_data do
        get ptr to fragment location;
        get fragment size;
        memcpy(buffer, fragment, size);
        flush_to_disk(buffer);
    end
    check inhibit(fragment);
end
```

The Data Writer is the element that interfaces with the local storage. The mechanism on which the data recording process works process is described in Algorithm 1. The most relevant elements in the initialization steps are the creation of a memory-aligned buffer (by using the *posix_memalign* function), starting the worker thread and setting the CPU affinity to the relevant physical cores. This last instruction is crucial to avoid loss in the total bandwidth. In addition, ensuring that the memory buffer is aligned is necessary when using the O_DIRECT flag. Initial tests showed that a 512 bytes alignment is not compatible with Micron® X100 drive and therefore a 4 KiB alignment was chosen.

An inhibit mechanism is also put in place in case the Data Writer is not able to sustain the rate of incoming data and, in this case, a warning message (e.g. *Dataflow is BUSY*) is issued. This means that, in order to avoid data loss, it is necessary to reduce the data extraction rate because the writing process is not able to keep up with the data production rate.

When a *trigger_flag* is enabled, data are sent to the Data Writer with a configurable request size and trigger rate, thus emulating a supernova burst event. Let S be the configurable request size and T the trigger rate. It follows that the data writing throughput to storage is given by the following equation:

$$\text{Throughput} = \text{S} \times \text{T}$$

Figure 7.13 shows the resulting throughput obtained with the prototype test application. Two MiniDAQ applications were started, each emulating the handling of 5 data links from a single readout unit. At present, the data writer is handled in one thread per application. Due to the substantial data sizes expected in the DUNE use case, two instances are sufficient to saturate the performance of the Micron® X100 storage drive. Tests were done by varying the data request size and measuring the writing throughput to the drive. The request rate was chosen to maximize the throughput in a stable system (no errors) and without any trigger inhibit. The maximum throughput from the Micron® drive is achieved starting from a data request size of 16 MiB. As a comparison, the synthetic performance test with two running threads has been included on the same plot (orange entries). Similarly to the results obtained in figure 7.10, the maximum throughput measured is approximately 8.5 GiB/s which almost fulfills the target value needed for the DUNE local storage buffer.

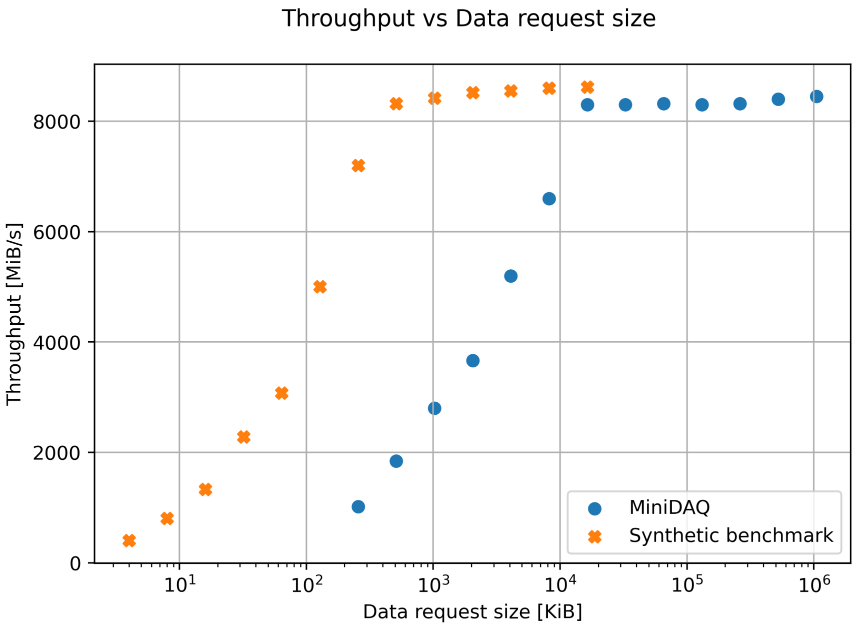


Figure 7.13: Comparison between the throughput achieved with a synthetic benchmark with 2 threads and with two MiniDAQ applications. The maximum bandwidth of the drive of approximately 8.5 GiB/s is achieved in both cases.

There is a notable difference in the measured throughput (figure 7.13) between the synthetic tests and the prototype of the more realistic data acquisition application: in the synthetic performance tests the buffers used are all memory aligned and multiples of the data request size.

In the MiniDAQ application, the data fragment size can vary, and it is not guaranteed to be a multiple of the writing I/O block size and not being memory aligned. Therefore, copying the data to a previously allocated buffer is necessary before flushing them to disk. This operation has a clear impact on the performance as seen in figure 7.13.

7.5.4 Discussion

The Micron® X100 device represents an interesting technology for the implementation of the DUNE local storage buffer. The maximum achieved throughput from the integration testing shows that the device is capable of continuously sustaining the incoming payload rate for the target range of 100 seconds. The Micron® X100 needs to use a block size of 16 MiB that is large compared to the raw wib frames of approximately 5568 bytes. Nonetheless, as shown from the integration testing this is not a critical factor as input data are memory-copied and can be flushed to disk with any block size providing that it is a multiple of the alignment factor. In fact, the writing bandwidth is saturated with larger block sizes in the case of X100 devices.

One potential drawback of the use of the Micron® X100 devices is that it uses 16 PCIe lanes for a single device. Therefore building a DUNE local storage buffer in a single readout unit with such devices would require at least 32 PCIe lanes for only the storage buffer (one device per CPU socket). This has to be added to the other PCIe devices (16 PCIe lanes for the FELIX I/O card and 16 PCIe lanes for the network interface card) that are also needed for each readout unit. Therefore, the total number of PCIe lanes that are needed for a single server is at least 96 PCIe lanes. Unfortunately, not many dual socket CPU servers available on the market offer such an amount of available PCIe lanes and, therefore, other solutions have to be found to mitigate this potential issue.

Finally, it is also worth noting that using an array of SSDs based on the PCIe Gen 4 can also provide the target throughput required for the DUNE local storage buffer. Preliminary results show that with four commercial SSDs in a RAID 0 configuration[6] it is possible to achieve a sustained throughput of 8.8 GiB/s for more than 100 seconds with a workload compatible with DUNE local buffer. Testing and integration of such a solution are being further explored by the DAQ team of the DUNE experiment. Table 7.4 shows a comparison between the maximum achieved sequential write throughput using DCPMMs, a single Micron® X100 device and RAID 0 con-

[6] RAID 0 (disk striping) is the process of dividing data into several data blocks (or stripes) and distributing them across multiple storage devices.

Table 7.4: Comparison of the maximum achieved sequential write throughput with the DUNE local buffer workload. Tests executed with three different classes of storage devices: DCPMMs, Micron® X100, RAID 0 with four SSD devices.

Device	**Max. sequential write throughput [GiB/s] DUNE local buffer workload**
DCPMMs	8.8
Micron® X100	8.5
RAID 0 configuration	8.8

figuration with four PCIe Gen 4 SSD devices[7]. The sequential write throughput for the RAID 0 configuration was obtained using the DUNE-DAQ software with the same workload expected for the DUNE local buffer: store data blocks of 5568 bytes at 166 kHz. The results show that the maximum achieved throughput between all the storage technologies is above 8.5 GiB/s. In particular, both DCPMMs and the RAID 0 configuration represent the ideal candidates for implementing the DUNE storage buffer because they manage to fully sustain the target rate for at least 100 seconds. Continuously monitoring the market trend of storage technologies is an essential task needed to evaluate candidate solutions that can fulfill the requirements of the local storage buffer.

7.6 Conclusion

This chapter showed the investigation, testing and integration of modern storage technologies that can be used as a potential storage hardware for the DUNE local buffer for supernova events. The uniqueness of the chapter is based on the fact that, at the time of this research, newly available storage devices have been evaluated using high-level applications that resemble the final use-case: a storage system that is capable of sustaining the raw data rate when a trigger signal is detected.

Persistent memory modules (DCPMMs) are memory-like devices that are capable of permanently storing data. They have been extensively tested in a synthetic environment as well as integrated with configurations relevant for the DUNE local storage buffer. Results have shown that the devices are capable of fully sustaining the target throughput of 8.8 GiB/s without any trigger inhibits or rate drop during their operation. The Micron® X100 SSD is a storage device that offers superior bandwidths compared to other SSD devices available on the market. From

[7] The SSD used for the investigation is a Seagate® FireCuda 1 TB device with a nominal maximum sequential write throughput of 4400 MB/s.

the benchmarking evaluation and from the integration testing the maximum achieved throughput that has been measured without any rate drop is approximately 8.5 GiB/s. This almost fulfills the target throughput required for the DUNE local storage buffer.

Although the DCPMMs and the Micron® X100 devices have been tested in different configurations the final results are always in line with the synthetic evaluations. In addition, being able to build an application based on COTS devices that is capable of sustaining the full raw data rate, either with an emulated environment or with the actual I/O workload, gives the confidence that there is no need to develop a custom solution for the DUNE local storage buffer. This also means that today's storage technologies can sustain the target throughput needed for the DUNE local buffer.

Summary of contributions:

- Complete benchmarking of persistent memory modules
- Testing of persistent memory modules with emulator for the ProtoDUNE data acquisition system
- Integration of persistent memory modules with the DUNE-DAQ software
- Synthetic benchmarking of Micron® X100 SSDs
- Integration with a prototype of the DUNE data acquisition system
- Testing novel PCIe Gen 4 SSD devices

8

Dataflow methods in particle physics experiments

The Dataflow System of both the ATLAS and DUNE experiments will need to sustain large incoming data bandwidths from the front-end electronics of the detectors. Both experiments, in fact, will need to extract the data from millions of channels. Typically, in the case of a collider experiment like ATLAS, events are formed hierarchically: successive trigger levels are used to evaluate if the data corresponding to a p-p bunch crossing represent an interesting signal. In the case of the DUNE experiment, the analog signals are continuously sampled at a fixed rate and left for the downstream DAQ system to decide if there is a local activity that is interesting and worth keeping. Due to the different readout approaches of the ATLAS and DUNE detectors, the Dataflow System of both experiments will need to be adapted accordingly to the various tasks of the system: buffer, format or transport requested data. This is done by combining local storage solutions and dataflow policies to orchestrate the flow of data effectively.

Emerging high-performance storage technologies are being used in the design of new distributed data acquisition system architectures where a large storage buffer decouples data production and data processing. This is motivated by the trend in the industry where large-scale computing systems are starting to decouple compute and storage capabilities [36] [37]. In fact, high-performance computing environments are experiencing a huge increase in data volumes which make the data consumption more difficult for the computing nodes. This is similar to the behaviour within the data acquisition system in which high I/O rates are expected in the readout and filtering nodes.

In the context of data acquisition systems for physics experiments, the decoupling of acquisition and processing is particularly interesting in those cases in which the acquisition rate may vary widely in time depending on the physical processes being measured: an intermediate storage element allows to dimension the data processing part of the system for an average load without needing to sustain temporary peaks. This chapter investigates different dataflow solutions and storage technologies tested and deployed within the ATLAS experiment for the design and development of the data acquisition system.

8.1 A distributed DAQ database (DAQDB)

The memory buffers in the readout nodes of both the ATLAS (Phase-II upgrade) and DUNE experiments can typically store only a few seconds of data due to the high incoming data rates. This is due to the capacity constraints and high cost of DRAM modules. As a result, the Data Selection systems of the experiments are tightly coupled to the data readout in order to select the most interesting events (ATLAS experiment) or to evaluate if there is an interesting activity in the detector (DUNE experiment). Traditionally, the data are transferred from the readout's buffers to the data selection nodes to complete the *physical event building* process. This is done by accessing, over the network, the data stored in the readout nodes and by building the aggregated information in a single continuous area in either DRAM memory or on storage media. However, this dataflow method is vulnerable to potential data loss due to network congestion or potential downstream issues during the data acquisition. For this reason, in the case of the Phase-II upgrade of the ATLAS experiment, a large distributed persistent buffering solution is designed to temporarily store all the incoming data from the readout system and serve fragments to the filtering nodes. Having a large storage buffer also provides less pressure on the network infrastructure and allows to decouple the readout from the data filtering, making the data acquisition system more resilient to failures.

By utilizing a large storage solution to temporarily hold the data a new dataflow approach becomes possible: *logical event building with hot storage*. This method consists in storing data fragments from the detector in a large distributed storage buffer like a key-value store[1] (KVS).

[1] A key-value store is a data structure that uses associative arrays for storing, retrieving and managing data entries

The event building process is then taken care of by performing only metadata operations with no data fragment transfer over the network. For example, event filtering nodes can request/delete fragments belonging to a specific event by only executing queries to the KVS.

DAQDB [38] is a distributed key-value store that implements the logical event building approach. It is an open-source project developed jointly by experts from CERN experiments and Intel® in CERN's OpenLab framework. The objective of the project is to provide a high-bandwidth (TB/s of throughput), generic data storage solution for data acquisition systems. This is done by taking advantage of modern storage media such as Intel® Optane™ devices previously discussed in chapter 7.

As part of this research work, the first sections of this chapter will be dedicated to the performance evaluation of the DAQDB system. The objective is to evaluate the system as an implementation of a large storage buffer (Storage Handler) of the ATLAS Dataflow system. This is also followed by the work done on the integration of DAQDB within the ATLAS data acquisition framework. The performance results as well as the experience in integrating a commercial-off-the-shelf (COTS) product like DAQDB may also be useful for the DUNE data acquisition system. Most of the material regarding the design of the DAQDB system can be found in [38], whereas a performance evaluation of the technology is described in [39].

8.1.1 High-level design of DAQDB

DAQDB is conceived as a KVS to be used in large, distributed and high-throughput DAQ systems like the ATLAS or DUNE experiments. Such experiments have, in fact, tight requirements: they need to sustain continuously terabytes per second of data, store hundreds of petabytes and be able to serve hundreds of nodes for data writing. For example, in the ATLAS experiment, the input request rate for the Phase-II system is 1 MHz distributed among 500 nodes and with an average fragment size of 10 KiB. In addition, the KVS has to provide read access to the stored fragment data to thousands of computing nodes. Therefore, the design of the DAQDB system has to include and address the tight requirements of the experiments from the beginning.

The DAQDB software provides a user-space[2] library with which a client node can push (or retrieve) fragments into (or from) the KVS. The *key* used in the DAQDB uniquely identifies a specific fragment. Its structure is configurable and it has been designed to be large enough to

(values). Each key is associated with only one value in the collection of entries.

[2] The term user-space refers to the portion of physical memory where a user can run processes with unpriviliged access, contrary to kernel-space where only OS kernel processes are executed with privileged access.

support specifics of DAQ systems: it has entries to identify the event, the sub-detector and the run of the experiment. In the current design, 40 bits are used for event ID, 8 bits for the sub-detector and 16 bits run IDs. The total length of the key is 64 bits. The DAQDB library also implements operations to access, insert or update entries into the data store, like `Get(key, options)`,`Put(key, value, options)`,`Update(key, value, options)`.

8.1.2 Architecture of DAQDB

DAQDB has been designed into a two-level buffering scheme. The objective of the first level buffer is to provide terascale storage capacity and be able to index the key-value pairs at high-bandwidth incoming rates. The underlying technology of the first-level buffering is based on the Intel® Optane™ DC Persistent Memory modules (DCPMMs) which provide data persistence. The current generation of DCPMMs provides up to 3TB of memory capacity per CPU socket[3]. Therefore, assuming a data stream at a rate of 100 Gbit/s, an input buffer holding a few minutes of data (per CPU socket) becomes feasible. Note that DAQDB has been optimized to work with persistent memory modules and therefore PMDK libraries [32] have been used extensively from the start of the design. The second level buffer is an optional storage layer that can be used to extend the storage capabilities provided by DAQDB in order to provide buffering capacity for hours of data taking. The underlying hardware is based on PCIe (NVMe) non-volatile storage media (SSDs) that is accessed directly from user-space.

DAQDB has been designed as a *generic high-throughput* solution for data acquisition systems of particle physics experiments. For example, in the case of the ATLAS experiment, the DAQ system for the Phase-II upgrade needs to sustain a higher write workload compared to the read workload. In fact, ATLAS will need to sustain approximately 5.2 TB/s in input writing and provide 2.7 TB/s of data reading. Therefore, the DAQDB data structure has been optimized for write workloads. In addition, the data produced by the readout nodes need to be stored at high rates. Therefore, a storage solution for the DAQ system needs to provide fast indexing capabilities. For this reason, DAQDB uses a modified Adaptive Radix Tree (ART) [40] structure. This structure has been carefully designed for efficient indexing in main memory and it provides O(k) time for accessing data surpassing lookup performance of highly tuned read-only search trees.

DAQDB is a *distributed* KVS which means that some of the data are stored on remote nodes. If

[3] This refers to the Intel® Optane™ DC Persistent Memory modules series 100

an operation has to be performed on remote nodes, DAQDB handles routing requests internally by implementing a *zero-hop* distributed hash table (DHT) [41]: each node in the network has enough routing information to directly route requests to the target nodes. With this method, latency is reduced because requests can be transmitted without routing through multiple nodes which is a typical problem in large-scale systems with hundreds of nodes. However, typically, the desired approach is to take advantage of data locality in order to reduce the network load and, therefore, readout nodes in the DAQ system can also be configured through DAQDB to act as storage elements.

Finally, another feature of the DAQDB architecture is the use of technologies that aim to reduce the CPU latency and the network overhead. In order to achieve this goal, *Remote Direct Memory Access* (RDMA) is used to access remote memory modules, in either DRAM or DCPMMs. In this way, it is possible to achieve a high-performance cluster that is capable of sustaining the target rates of particle physics experiments. However, the DAQDB system needs to provide read access to potentially thousands of clients (in parallel) which is not feasible with RDMA-capable network interface cards (NICs) because they require caching for each connection state. In order to achieve this goal, DAQDB uses eRPC [42] as the transport layer in order to scale to a large number of nodes.

Deployment modes

The DAQDB software library can be deployed into two modes of operation based on the location of the storage nodes. Figure 8.1 shows the base deployment mode in which the computing nodes are used for both detector readout applications as well as for storage (both DAQDB buffering levels). Running both workloads into the same node allows to store data locally and, therefore, reducing the network load. Subsequently, a data selection application can then request the data by using the DAQDB library to route the requests from the appropriate readout nodes. Figure 8.2 shows a different deployment approach, where a specialized set of nodes are used for only the storage functionality (one or two levels of buffering). Both readout (write) requests and data selection (read) requests are routed internally via the DAQDB interface to the storage nodes which run a standalone application known as DAQDB *thin-server*. The advantage of this mode of operation is that storage and data processing workloads are executed on different servers. However, more nodes are needed to provide the same amount of storage compared to the base deployment mode.

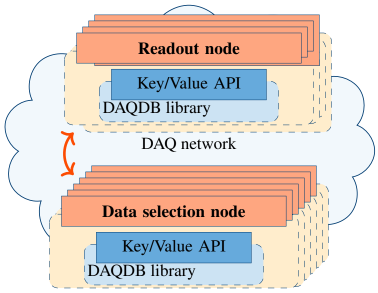


Figure 8.1: Base deployment mode.

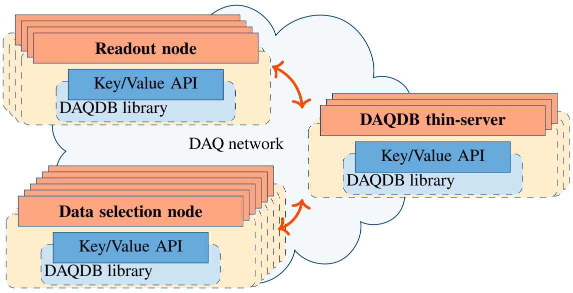


Figure 8.2: Deployment mode with separate storage.

8.1.3 Performance evaluation

The performance evaluation of DAQDB was done to decide whether the technology is suitable for the data acquisition system of the ATLAS experiment. The testing approach relies on characterizing the writing access pattern with both the local storage (local testing) and with the DAQDB remote thin server (remote testing). Specifically, this approach has been selected in order to investigate whether DAQDB is capable of writing data fragments into the KVS at the rates and at the scale required for the Phase-II upgrade of the ATLAS experiment. The local storage testing provides insights whether the KVS is capable of indexing the data fragments at the target rate and the remote testing provides the possibility of evaluating the DAQDB technology in a more realistic configuration suitable for the ATLAS experiment where readout nodes send the data over the network to the Storage Handler nodes (Chapter 6). Subsequently, based on the benchmarking results, the reading access pattern is also evaluated. Special focus is given to the remote testing because it provides the opportunity to investigate whether the storage nodes (DAQDB thin servers) are capable of providing fragments to the relevant data selection nodes at the target rates needed by the ATLAS experiment.

Testing environment

The performance evaluation of the DAQDB system was done using four servers equipped with a Intel® Xeon® Platinum 8280L 28-core dual-socket CPU (Cascade Lake) running at 2.7 GHz. The platform is also equipped with a total of 3 TB of DCPMMs and a total of 96 GB of DDR4 DRAM volatile memory. The servers are connected with Mellanox ConnectX-5 network cards providing a 100 Gbps connectivity between the servers. Finally, note that the testing was performed using only one instance of DAQDB running on the servers in order to avoid potential cross-NUMA interactions between the two CPU sockets within the servers. Table 8.1 summarizes the specification of one of the machine nodes used for the evaluation of DAQDB.

Table 8.1: Overview of the test machine used for the DAQDB evaluation.

CPU	Intel® Xeon® Platinum 8280L 2.70 GHz (Cascade Lake), dual socket L1d cache 32K L2 cache 1024K L3 cache 3942K
DRAM	DDR4 DRAM 16 GB, 2666 MT/s, 12 slots Product number: Kingston KSM26RS4
DCPMM	DDR-T 512 GB, 2666 MT/s, 12 slots Product number: Intel®NMA1XBD512GQS
OS	CentOS 7, Linux Kernel 4.15.0
SW	ipmctl v.01.00, PMDK v.1.9
Network	Mellanox ConnectX-5

Single node performance

The first performance evaluation of DAQDB was executed in a simple scenario consisting of a single node and with a single thread. This was achieved by deploying a write thread that is emulating the workload of a readout application and that is executing *put* requests with a predefined block size. The results obtained for DAQDB are compared with Redis [43] which is a popular and widely-used in-memory KVS. The main difference between DAQDB and Redis is that the keys and the values used for the data store can be permanently stored in DAQDB, whereas this is not possible in Redis as data (both keys and values) reside inside the DRAM.

Figure 8.3 shows the rate of put requests as a function of the I/O block size when writing into both the DAQDB and the Redis key-value stores. The selected I/O block sizes range from 256 bytes up to 8192 bytes because they represent the ideal size of an ATLAS detector fragment. The comparison illustrates that DAQDB achieves a mean rate of put requests just below 300 kOPS in the entire range of I/O block sizes, and that is around three times higher than Redis. This initial benchmark suggests that DAQDB is capable of efficiently allocating and indexing the incoming requests into persistent memory as compared to Redis. Note that the results for the Redis system are obtained using *redis-benchmark* which is a utility used to benchmark a Redis instance by emulating a custom workload. Specifically, different block sizes were used to measure the maximum achieved rate of insert operations. In addition, the Redis key-value store was configured to work with persistent memory modules (DCPMMs) and non-volatile memory in order to compare the data injection rates with DAQDB.

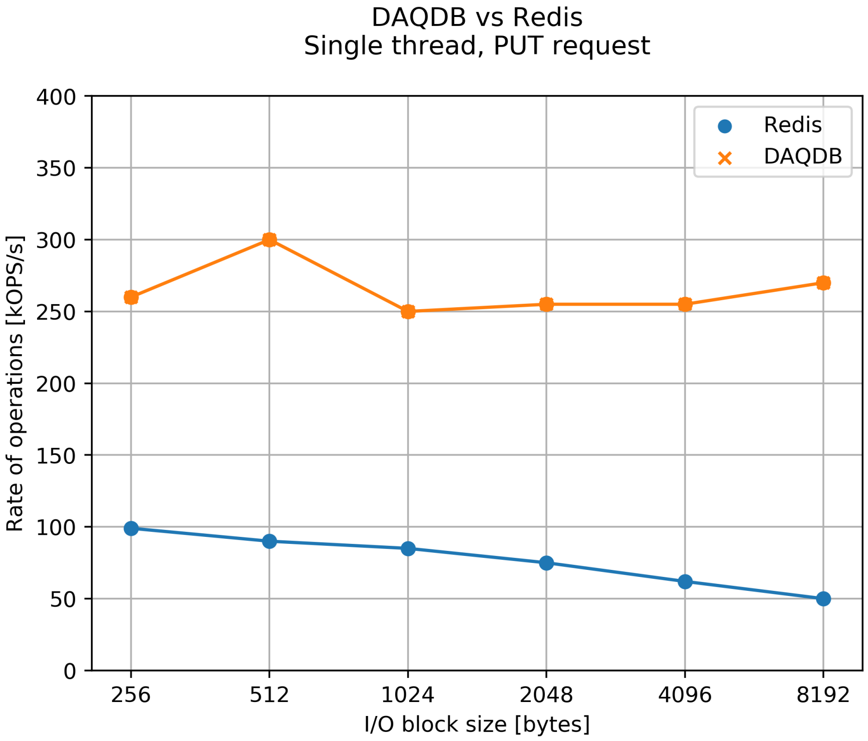


Figure 8.3: Comparison of the rate of put requests between DAQDB and Redis. Results expressed in operations per second.

Local storage evaluation

The next step in the evaluation of DAQDB consists in the performance assessment of the system in a more realistic scenario. This is done in view of the possible application of DAQDB in the data acquisition system of the ATLAS experiment. The ATLAS data acquisition system takes advantage of the several CPU cores provided in the computing infrastructure to deploy its readout and event filtering processes. Therefore, it is important to investigate the behaviour of DAQDB as a function of the number of threads and as a function of the requested block size. Figures 8.4 illustrates the total throughput as a function of the number of executing threads when inserting (PUT requests) fragment data into the DAQDB key-value store. Similarly, figure 8.5 shows the average throughput when retrieving (GET requests) fragment data. The results were obtained running the executing threads locally (i.e. no data transfer over the network) and with two block sizes: 1 KiB and 10 KiB. Note that the maximum number of threads used in the evaluation is 28 because it represents the total number of physical cores available on a single CPU socket in the testing machine. In addition, as an example, writing measurements were done executing one or several threads injecting fixed I/O block sizes into DAQDB. The total throughput is obtained by summing each thread's performance.

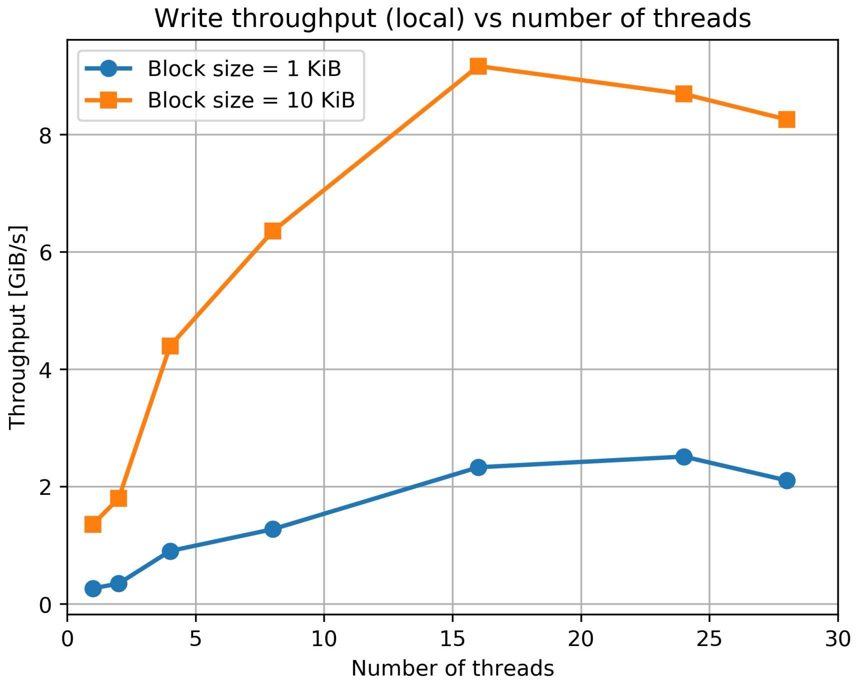


Figure 8.4: Average write throughput as a function of the number of threads for a block size of 1 KiB and 10 KiB. Results obtained when using DAQDB locally.

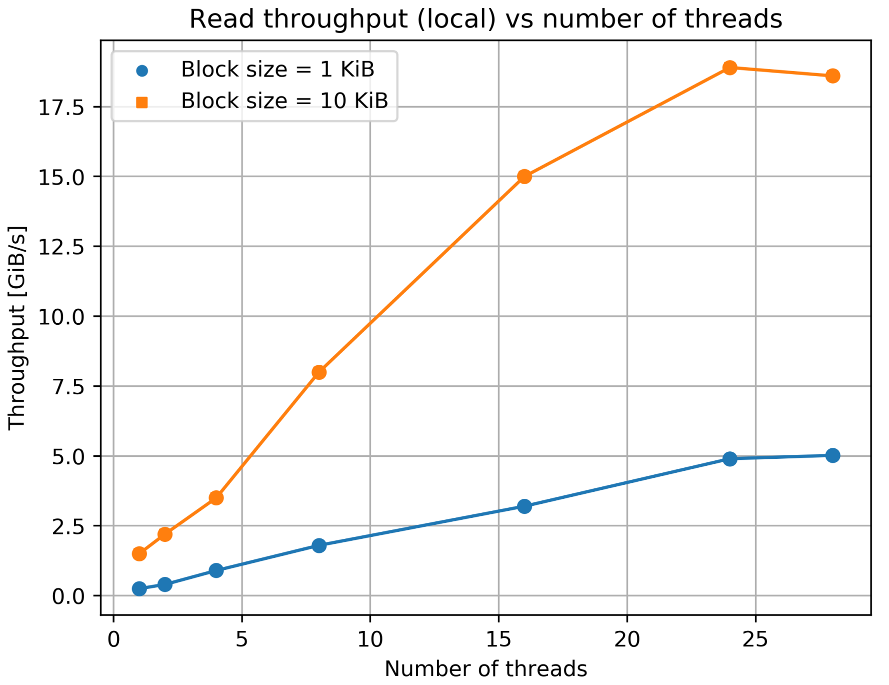


Figure 8.5: Average read throughput as a function of the number of threads for a block size of 1 KiB and 10 KiB. Results obtained when using DAQDB locally.

In the case of the local write throughput (figure 8.4), DAQDB saturates the bandwidth at approximately 8.5 GiB/s with a block size of 10 KiB and starting from 16 threads. It is interesting to note that this limit is close to the maximum throughput achieved with the DCPMMs as discussed in the benchmarking results in chapter 7. In fact, as mentioned earlier, DAQDB uses the DCPMMs as the storage media for the first-level buffering which also constitute the bottleneck when performing local operations on the KVS. In the case of the 1 KiB block size the bandwidth is saturated at around 2 GiB/s. On the other hand, local read requests (figure 8.5) show that the throughput reaches a maximum value just below 20 GiB/s when issuing 28 threads and with a block size of 10 KiB. Results for the read throughput are in line with the literature results described in [29] and with the benchmarking measurements for the DCPMMs illustrated in figure 7.4.

The local testing of DAQDB shows satisfactory results for the Phase-II upgrade of the data acquisition system of the ATLAS experiment. As illustrated in chapter 6, the target write bandwidth that the ATLAS storage system will need to sustain is approximately 20 GB/s for writing and 1 GB/s for reading across 300 dual socket CPU servers (Storage Handler nodes). Running DAQDB in local mode shows that, per CPU socket, the achieved bandwidth for reading data are perfectly satisfied whereas the achieved maximum write bandwidth of 8.5 GiB/s is close to the target required by the ATLAS experiment for the Phase-II upgrade. However, it is important to note that these tests neglect some of the DAQDB features like offload to the second-line buffering system or features that, in practice, are needed for the data acquisition system like data clean up after reading. These features were not fully deployed into DAQDB at the time of writing and, therefore, they were not included in the testing evaluation.

Remote storage evaluation

As a reminder, for the Phase-II upgrade the readout system of the ATLAS experiment will require approximately 5 TB/s of input bandwidth from 500 nodes and distributed across a storage system of approximately 300 servers (Storage Handler nodes). If DAQDB is to be used as the underlying technology for storing and retrieving the data fragments, it has to be deployed as a thin-server. In this way, Storage Handler nodes act as the storage elements, whereas both Readout and Event Filter nodes act as DAQDB clients. Therefore, it is important to evaluate the performance of DAQDB when writing and reading requests are done across different nodes over the network.

Figure 8.6 and figure 8.7 illustrate, respectively, the remote write and read throughput as a function of the number of threads for block sizes of 1 KiB and 10 KiB. The maximum achieved write throughput is approximately 1.5 GiB/s for a block size of 1 KiB and 3.5 GiB/s for a block size 10 KiB. In the case of remote reading, the maximum throughput obtained with 28 threads is approximately 1.5 GiB/s (block size of 1 KiB) and 6.5 GiB/s (block size of 10 KiB). Overall, the remote testing shows that the write and read throughput is, respectively, two to three times lower when DAQDB is deployed as a thin-server (remote mode) compared to the local results. The origin for such low values is two-fold. One reason is due to use of the eRPC transport layer between different nodes. In fact, eRPC uses blocks of 1000 bytes as the maximum size unit (MTU) for the network transmission. This means that writing or reading I/O block sizes bigger than the MTU value requires more data transmission and more CPU time to process the data, effectively making the server CPU-bound (case for a block size of 10 KiB). Another reason for the difference in throughput when running locally compared to running in remote mode is that in the latter case threads need to be synchronized, as of the current implementation of DAQDB. Therefore, when running in remote mode, executing threads need to wait until a response is received before proceeding with the workflow and, as a consequence, this may reduce the overall throughput (client becomes latency-bound).

8.1.4 Integration with the ATLAS TDAQ framework

A lot of effort was dedicated to the integration of the DAQDB technology with the ATLAS TDAQ framework. The objective of this task was to investigate how DAQDB can be operated and deployed in a large software framework like the one used in the ATLAS experiment. As discussed in [39], the integration approach consisted in developing standalone *put* and *get* applications that would act as readout and event filtering nodes. Such applications were configured to use DAQDB as the backend storage. This flexible approach allows to benchmark and validate the system with applications that have been extensively tested with the synthetic benchmarks.

Many difficulties were encountered during the integration process. DAQDB was deployed on only one CPU socket and it was not trivial to set up all the executing threads to the correct physical cores. Note that when using the full TDAQ software suite, many threads used for the control, configuration and monitoring are also started. In addition, the offloading of data from DCPMMs to SSDs needed to be disabled because read threads were crashing when looking for data hosted on the second level buffering scheme. The offloading was, in fact, a recently introduced feature that was not completely integrated with the DAQDB key-value store. Overall, the

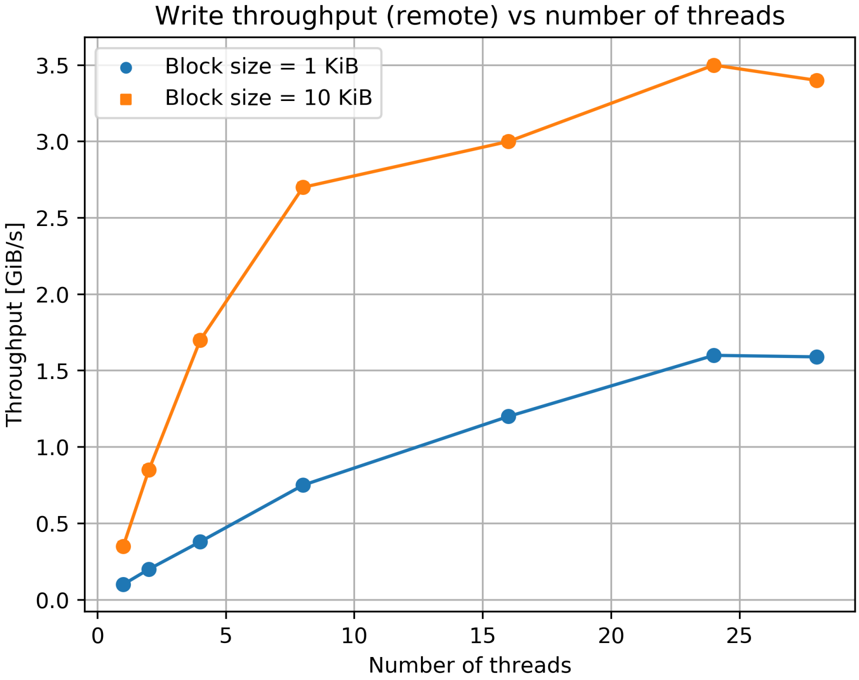


Figure 8.6: Average write throughput as a function of the number of threads for a block size of 1 KiB and 10 KiB. Results obtained when using DAQDB servers in thin mode.

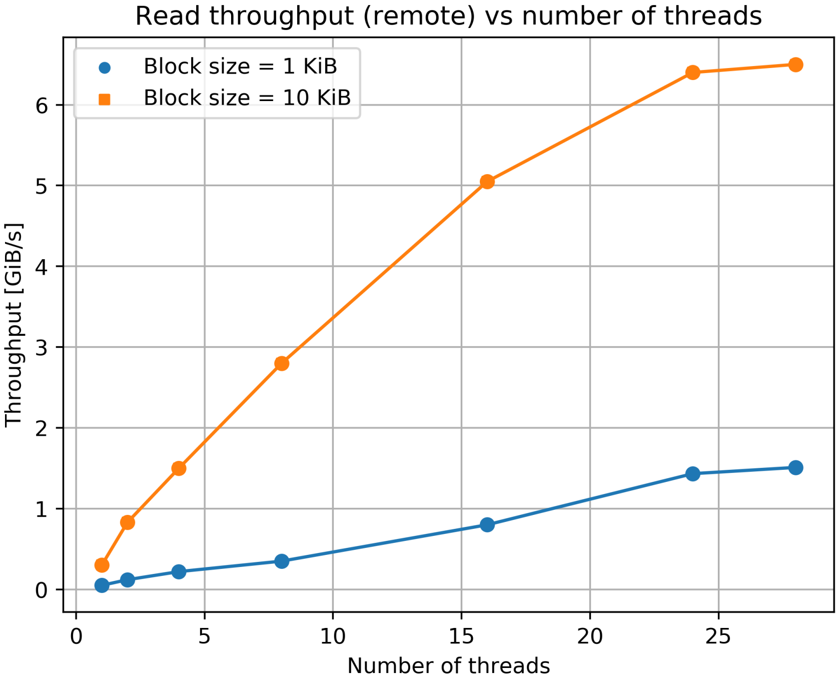
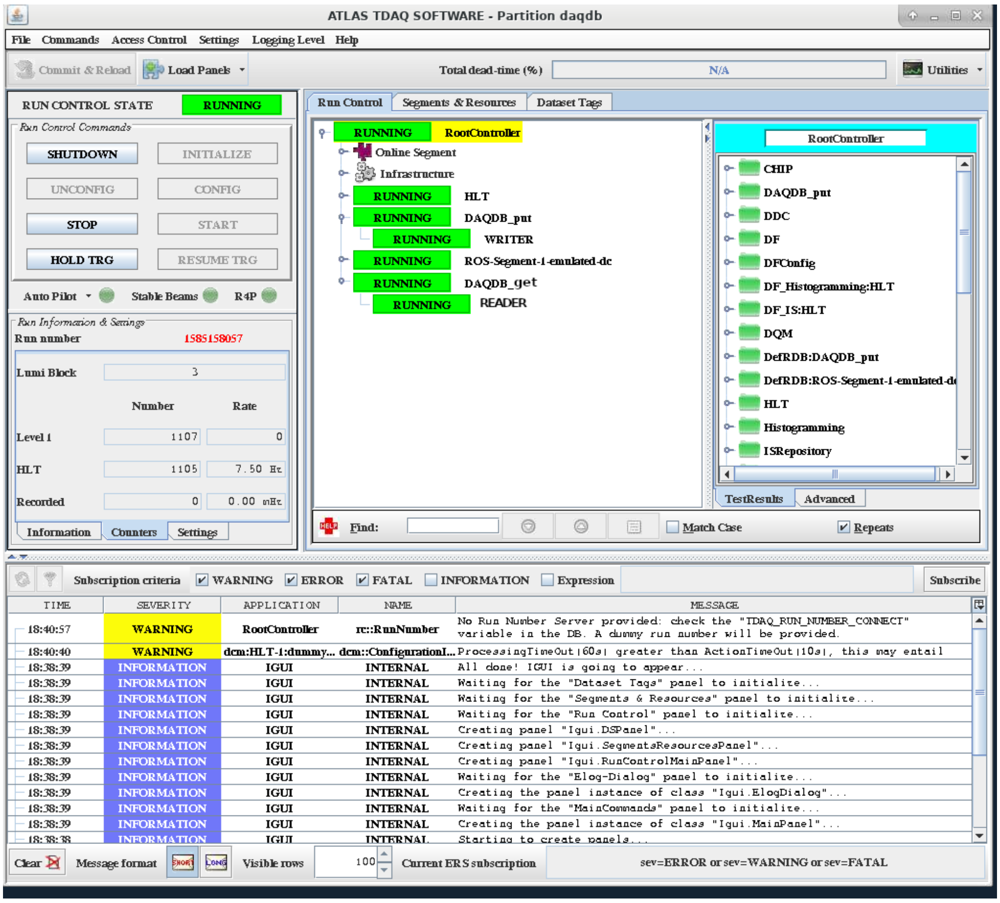


Figure 8.7: Average read throughput as a function of the number of threads for a block size of 1 KiB and 10 KiB. Results obtained when using DAQDB servers in thin mode.



RUN CONTROL STATE: RUNNING

Total dead-time (%): N/A

Run Information & Settings

Run number: 1585158057

Lumi Block: 3

	Number	Rate
Level 1	1107	0
HLT	1105	7.50 Hz
Recorded	0	0.00 mHz

- RUNNING RootController
  - Online Segment
  - Infrastructure
  - RUNNING HLT
  - RUNNING DAQDB_put
    - RUNNING WRITER
  - RUNNING ROS-Segment-1-emulated-dc
  - RUNNING DAQDB_get
    - RUNNING READER

RootController

- CHIP
- DAQDB_put
- DDC
- DF
- DFConfig
- DF_Histogramming:HLT
- DF_IS:HLT
- DQM
- DefRDB:DAQDB_put
- DefRDB:ROS-Segment-1-emulated-d
- HLT
- Histogramming
- ISRepository

☐ Match Case ☑ Repeats

Subscription criteria ☑ WARNING ☑ ERROR ☑ FATAL ☐ INFORMATION ☐ Expression

TIME	SEVERITY	APPLICATION	NAME	MESSAGE
18:40:57	WARNING	RootController	rc::RunNumber	No Run Number Server provided: check the "TDAQ_RUN_NUMBER_CONNECT" variable in the DB. A dummy run number will be provided.
18:40:40	WARNING	dcm:HLT-1:dummy...	dcm::ConfigurationI...	ProcessingTimeOut|60s| greater than ActionTimeOut|10s|, this may entail
18:38:39	INFORMATION	IGUI	INTERNAL	All done! IGUI is going to appear...
18:38:39	INFORMATION	IGUI	INTERNAL	Waiting for the "Dataset Tags" panel to initialize...
18:38:39	INFORMATION	IGUI	INTERNAL	Waiting for the "Segments & Resources" panel to initialize...
18:38:39	INFORMATION	IGUI	INTERNAL	Waiting for the "Run Control" panel to initialize...
18:38:39	INFORMATION	IGUI	INTERNAL	Creating panel "Igui.DSPanel"...
18:38:39	INFORMATION	IGUI	INTERNAL	Creating panel "Igui.SegmentsResourcesPanel"...
18:38:39	INFORMATION	IGUI	INTERNAL	Creating panel "Igui.RunControlMainPanel"...
18:38:39	INFORMATION	IGUI	INTERNAL	Waiting for the "Elog-Dialog" panel to initialize...
18:38:39	INFORMATION	IGUI	INTERNAL	Creating the panel instance of class "Igui.ElogDialog"...
18:38:39	INFORMATION	IGUI	INTERNAL	Waiting for the "MainCommands" panel to initialize...
18:38:39	INFORMATION	IGUI	INTERNAL	Creating the panel instance of class "Igui.MainPanel"...
18:38:38	INFORMATION	IGUI	INTERNAL	Starting to create panels...

Visible rows: 100

Current ERS subscription: sev=ERROR or sev=WARNING or sev=FATAL

Figure 8.8: Image of the ATLAS Run Control software with DAQDB *put* and *get* applications.

integration process of DAQDB within the TDAQ framework proved to be less straightforward than anticipated initially.

The integration experience proved to be very useful in understanding how challenging it is to integrate a COTS solution into a data acquisition system of a running experiment. Although many drawbacks were experienced, DAQDB was eventually integrated within the ATLAS TDAQ framework and tested with both local writing and remote reading. Performance results showed the same throughput values obtained in the previous sections. Figure 8.8 shows an image of the ATLAS TDAQ Run Control application software with the integrated DAQDB writer (*put*) and reader (*get*) applications.

8.1.5 Discussion

The performance evaluation of DAQDB gives promising results for the use of the technology as a generic KVS solution for data acquisition systems. The throughput obtained when testing the system in local mode shows that DAQDB is capable of sustaining the target write and read

Table 8.2: Code profiling summary when executing a write thread (PUT request) with a local DAQDB instance and with I/O block sizes of 100 bytes and 10 KiB.

Function	Time spent with block size of 100 B	Time spent with block size of 10 KiB
→ DaqDB::KVStore::Alloc	77 %	18 %
↳ DaqDB::TreeImpl::findValueInNode	—-71%	—-11%
↳ pmemobj_mutex_lock	——— <33%	——— <5%
↳ pmemobj_mutex_unlock	——— <28%	——— <5%
→ pmem_memcpy_nodrain	<5%	65%
→ DaqDB::KVStore::Put	5%	12%

throughput required by the ATLAS experiment for the Phase-II upgrade. In particular, DAQDB showed to be a promising solution that is capable of indexing and retrieving data at high rates. However, tests were done locally and were not performed with both the read and write access pattern running in parallel on DAQDB. This is a feature that in practice is needed when running inside the data acquisition system.

In addition, the size of the data fragments have an impact on the persistent memory bandwidth and on the access latency. Table 8.2 shows the summary of the code profiling, i.e. average time spent for a specific instruction, when executing a local write thread on DAQDB. Results are shown for a small block size of 100 bytes and a block size of 10 KiB. As it can be noted, with small fragments most of the time is spent allocating keys and traversing the DAQDB ART tree data structure which therefore becomes the bottleneck for small data objects. However, for bigger block sizes such as 10 KiB, most of the time is spent storing the values inside the persistent memory modules (*pmem_memcpy_nodrain* instruction).

Deploying DAQDB in local mode is not sufficient for large scale data acquisition systems like in ATLAS because hundreds of nodes are needed to reach the target bandwidths expected for the Phase-II upgrade. In practical terms, DAQDB needs to be deployed as a thin-server in remote mode. However, the throughput obtained in this case suffers from network issues due to the use of the eRPC transport protocol and synchronization problems between threads. For this reason, because DAQDB lacks a lot of work from the network side and it still misses a few features relevant for DAQ systems - being able to run seamlessly both readout and event filtering threads - the development effort of the project was merged with a similar and more mature storage solution: DAOS object store [44]. This is a solution that is currently being investigated by the ATLAS DAQ team at CERN.

8.2 Development of the ATLAS Dataflow System

As discussed in chapter 6, the Storage Handler of the ATLAS Dataflow system will need to sustain a throughput of approximately 5.2 TB/s in input from 500 Data Handler nodes and 2.7 TB/s in output to serve requests from 3000 event filter nodes. Each fragment can be requested individually by the Event Filter and, therefore, must be software-addressable. Generic solutions such as distributed file systems like Lustre [45] or object stores like Ceph [46] are used for indexing data across a distributed network of hundreds of nodes exists. Such solutions implement data indexing with different mechanism:

1. Data indexing with a server used to keep a global index: such solution may suffer from scalability, bottlenecks because of using a centralized system.
2. Data indexing using an algorithm to compute the location of the entries based on the properties of the data: such solution may suffer from flexibility or fault-tolerance issues.

In [47], the author has evaluated the performance of distributed file systems and showed that such solutions do not provide all the requirements needed for the ATLAS data acquisition system. Other solutions include the use of distributed key-value stores such as the previously mentioned Intel® solutions based on the persistent memory technology: DAQDB and DAOS. An alternative approach is to take advantage of the static nature of the Dataflow system and develop a solution that combines an indexing data structure to place the fragments across the nodes and a local indexing structure for organizing the fragments inside the storage media. The following sections show an alternative method of developing the ATLAS Dataflow system with a combination of both COTS solutions and ad-hoc applications.

8.2.1 Global indexing

The ATLAS Dataflow system has a static organization: the topology never changes during a data-taking session (i.e. the run number of the experiment) and a specific fragment always comes from the same subset of readout nodes. Similarly to what has been developed for DAQDB, a fragment is divided into three identifiers (event identifier, sub-detector identifier and the run number of the experiment) that are used for both global and local indexing.

At the global level, a fragment is assigned (or retrieved) to (or from) a server node by applying modulo operations and static rules, using the fragment identifiers and the topology of the data-

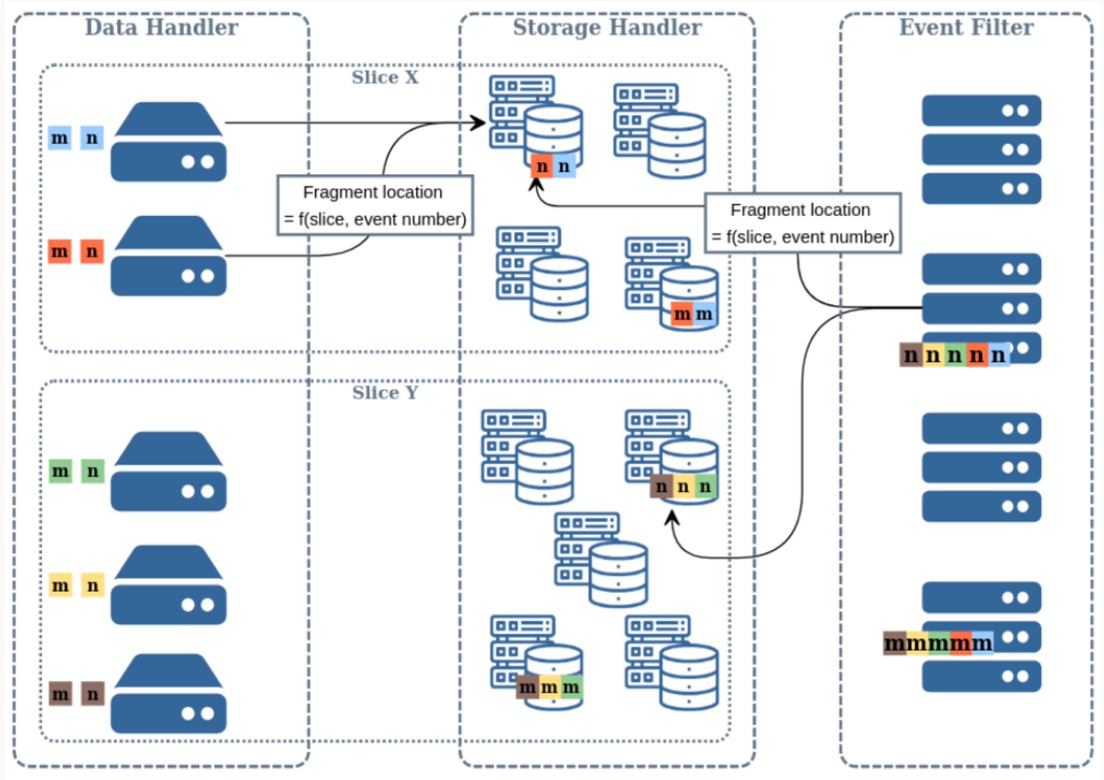


Figure 8.9: Diagram illustrating the placement algorithm for the global indexing method. Fragments from the different detectors are illustrated with different colors and they are grouped into different slices. Fragments corresponding to the same event (e.g. "n" or "m") are sent to the same node within the selected slice. The location of the fragments is computed by taking into account the event number and the slice identifier.

taking session. In practice, a subset of nodes is returned instead of a single server in order to handle failures and make the overall system more fault tolerant. Note that, although not tested, erasure coding is envisioned to handle data protection. Such global placement algorithm is used by both Readout nodes and Event Filter nodes to determine the location of the fragments. The advantage of using a global placement algorithm is that there is minimal communication between nodes and no central resource is needed. Figure 8.9 illustrates the placement algorithm of the global indexing method where fragments from the detectors are illustrated with different colors and grouped into several Dataflow slices. Given a Dataflow slice, the fragments hosted by the Data Handler are assigned to a subset of Storage Handler nodes by taking into account only the slice identifier and the event number: fragments corresponding to the same event are assigned to the same node. For example, fragments corresponding to event "n" are assigned to the same Storage Handler node within the Dataflow slice. Similarly, an Event Filter node can request all the fragments corresponding to a single event by computing the fragment location using the slice number and the event number.

8.2.2 Local storage management

At the local level, fragments must be allocated, retrieved, deleted, and available storage space from multiple drives must be managed. These tasks are usually performed by file systems which use data structures (e.g. Adaptive Radix tree, Log-structured merge-trees, Hash Tables) for providing indexed access to files. An example of a widely-used[4] file system is ext4. In ext4, data entries (i.e. files to write) are split into blocks, usually multiples of a 4 KiB block size, and stored ideally in a continuous location inside the storage media (e.g. hard-disk drive, SSDs, etc.). At the beginning of each data block, the file system also writes the metadata information (file type, access permissions, file size, location and length of data blocks) for the stored file and keeps track of all the blocks for each file. Ext4 uses also a delayed allocation strategy to allow the file system to collect the data being written to the disk before allocating space to it. This helps making sure that the data space will be contiguous for easier lookup. File systems represent a standard tool for managing data across storage media, however a full evaluation of file systems goes beyond the scope of this research. More details on the inner mechanisms and the features of the ext4 file system are described in [48].

Another approach for the storage management at the node level is to use a local key-value store. RocksDB[49] is a COTS-based high-performance key-value store optimized for low latency storage media. Contrary to DAQDB which is a distributed key-value store designed to support hundreds of nodes, RocksDB implements the storage management locally at the single node level. At the core of its design, RocksDB relies on three basic elements: memtable, SST (Sorted String Table) files and log files. The memtable is an in-memory data structure to store the key-value pairs of the data entries. The log file (also known as write-ahead-log or WAL) is a file that is used to keep track of the data entries that are still in memory. The log information can also be temporarily stored in a cache buffer in the physical memory before being written to the WAL. Finally, when the memtable reaches its maximum size, its contents are flushed to the SST files and stored on persistent storage media. SST files persistently store the key-value pairs in an organized sequence of levels starting from Level-0. When one level reaches its maximum size limit, then the SST file is merged with the files in the next level that have overlapping key-ranges. This process is known as compaction mechanism and it allows to control the disk writes on the underlying the storage media. Figure 8.10 illustrates the basic architecture of RocksDB, highlighting memtables, the SST files and the WAL log file which represent the

[4] Default Linux file system since 2010.

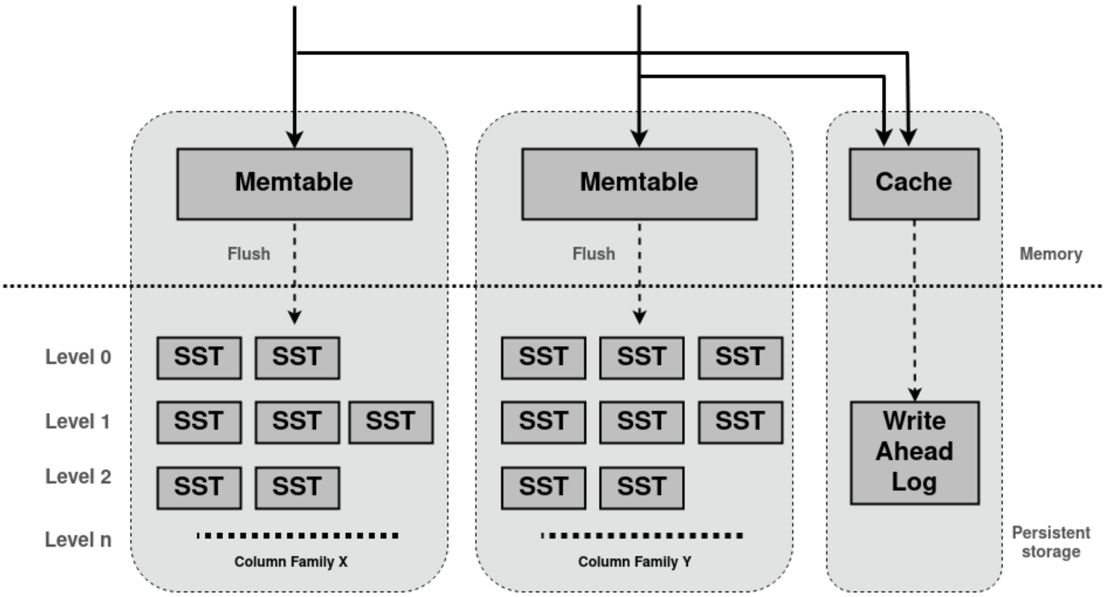


Figure 8.10: Diagram illustrating the basic architecture of RocksDB.

three core elements of the architecture. The figure also illustrates which components reside in physical memory (memtable, cache) and which ones on persistent storage (SST files, WAL). Finally, figure 8.10 also shows the logical partition of the key-value store in different columns, each with its own memtable buffer.

8.2.3 Evaluation of local storage management solutions

In the context of the data acquisition system, RocksDB and the ext4 file system were tested as a potential solution to develop the local storage management feature needed for storing ATLAS fragments. The objective of the evaluation was to measure the write throughput for the ext4 file system (ext4) and for the RocksDB key-value store. This was performed for different object sizes. The testing consisted in measuring the write throughput when inserting data objects with a specific value size into RocksDB or when writing to a file on an ext4 system. All tests were done on the same SSD drive (Intel® Optane™ P4800X, 375 GB).

Once a data fragment reaches a Storage Handler server, the underlying local storage management technology needs to allocate and flush the data. The maximum write throughput of any local storage solution cannot exceed the bandwidth provided by the underlying storage media. Therefore, the selected storage solution needs to provide a write bandwidth as close as possible to the storage devices. Figure 8.11 shows the write throughput as a function of the block size for RocksDB ranging from 4 KiB up to 1 MiB. When testing RocksDB with the default configuration options (throughput without optimizations) the throughput reaches a maximum

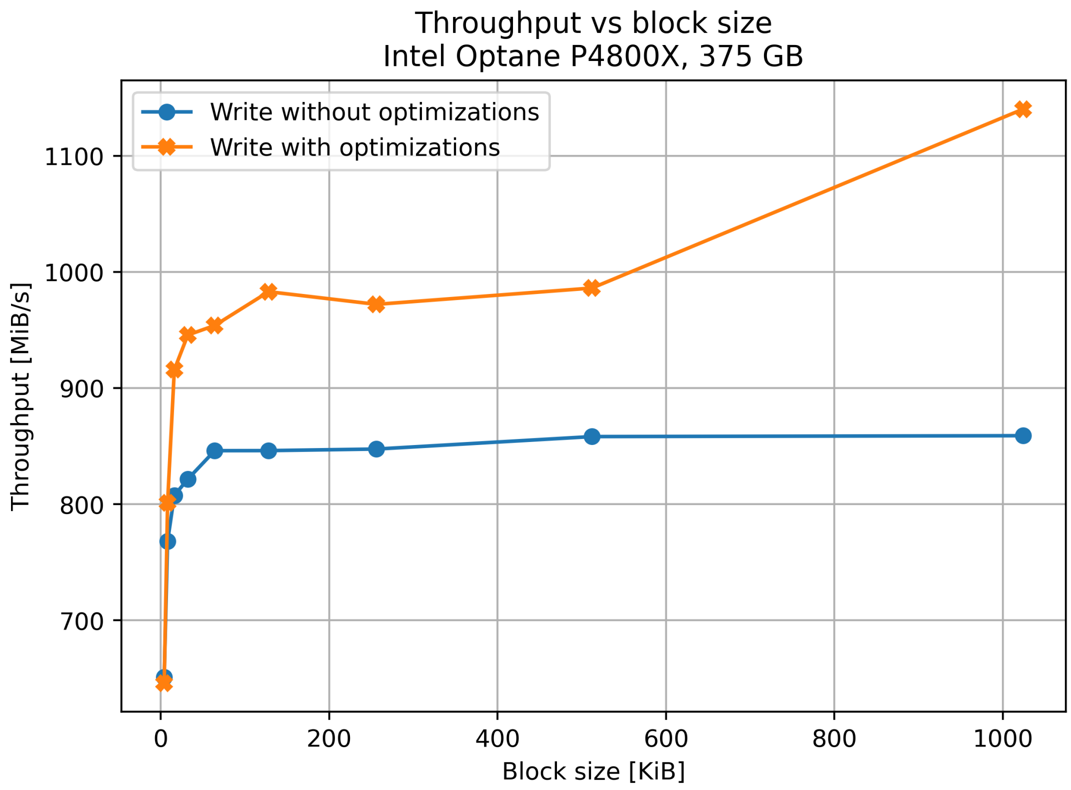


Figure 8.11: RocksDB write throughput as a function of block size. Comparison of the system with and without optimizations.

value of approximately 850 MiB/s which is less than half of the total write bandwidth provided by the SSD. In fact, the maximum write throughput of the SSD device used for the testing is 2 GiB/s. Therefore, the RocksDB key-value store needs to be properly tuned for the workload of the ATLAS data acquisition system.

In order to achieve the highest write bandwidth from RocksDB the Write Amplification factor needs to be reduced as much as possible. This factor represents the ratio of bytes written to the storage device with respect to the bytes written to the key-value store. As an example, a Write Amplification factor of ten means that the observed writing throughput to RocksDB is one tenth of the total bandwidth provided by the storage device. By changing the default configuration options it is possible to tune RocksDB to the desired workload and achieve lower Write Amplification factors. This was achieved by tuning following parameters:

- Asynchronous writes

  Enabling asynchronous write makes sure that each the transfer from the physical memory to the underlying storage media (write operation) happens asynchronously in RocksDB.

- Disable Write-ahead logging (WAL)

  By disabling WAL, it is possible to avoid spending time and CPU cycles in syncing the data files. Instead, it is possible to sync manually once the data flushing operation to disk is finished. The risk of disabling WAL is that the atomicity in the key-value store in case of multiple concurrent write operations may be compromised: in case of errors during the write operations the recovery of the database is not guaranteed.

- Enable direct I/O

  Enabling direct I/O when flushing to storage disk provides higher throughput as the operation is done bypassing the OS cache. In fact, when using buffered I/O the data is copied between storage and memory because of the OS page cache.

- Enable *WriteBatch*

  Enabling *WriteBatch* provides atomic edits to the key-value store by grouping a series of updates in a batch to be committed.

- Disable compression in the SST files

  Disabling the compression on the SST files allows the writing thread to perform only the flushing to storage media.

- Increase parallelism

  By default, RocksDB uses only one background thread for flush and compaction. By increasing the parallelism, RocksDB will automatically use the total number of physical cores for the flush and compaction operations.

In addition, other minor options were also tuned following the RocksDB Tuning Guide[5]. Figure 8.11 shows the resulting throughput as a function of the block size with the performance tuning for the write workload. As it can be noted, in this case the maximum throughput achieved with RocksDB reaches at a block size of 1 MiB a maximum value above 1100 MiB/s. This is a 30% gain compared to the default options. The resulting Write Amplification factor for a block size of 1 MiB is 1.8, compared to the previously obtained 2.34 when RocksDB was not tuned. Although the performance gain is remarkable, the achieved throughput is still too low compared

[5] Available online on the Github page of RocksDB

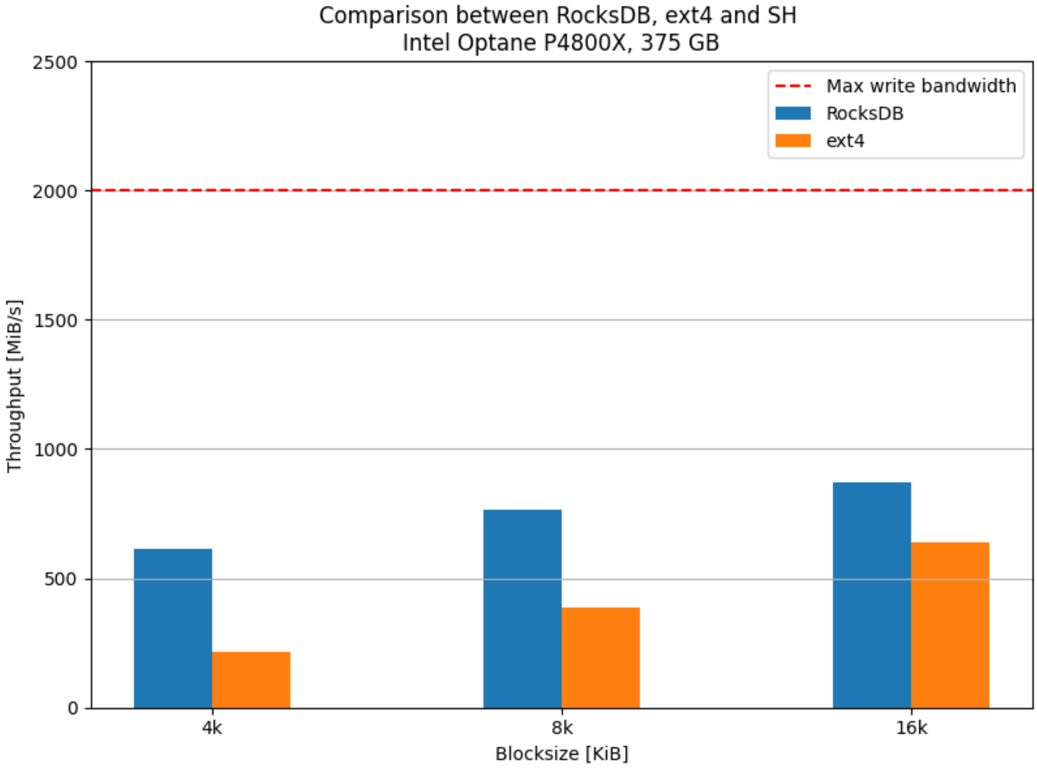


Figure 8.12: Write throughput comparison between RocksDB key-value store and the ext4 file system.

to the maximum bandwidth provided by the storage disk. Note that RocksDB was also tested for a block size of 100 MiB and the achieved throughput is still saturated at approximately 1100 MiB/s.

Figure 8.12 shows a comparison of the write throughput between RocksDB and the ext4 file system for three block sizes relevant for the ATLAS data acquisition system. In fact, as mentioned in chapter 6 the average fragment size is 10 KiB and therefore it is interesting to investigate the bandwidth for block sizes that are in the range of the expected ATLAS fragments. Figure 8.12 shows that the ext4 file system suffers from many overheads (e.g provide concurrent access, data safety, atomicity) which results in a low write throughput, making the solution unusable. The RocksDB store outperforms the ext4 file system solution by efficiently limiting the amount of overheads for the specific application. Nonetheless, the resulting throughput is only 50% of the nominal bandwidth of the drive.

To solve the high Write Amplification factors of RocksDB and the ext4 file system a custom object store (Storage Handler object store) was developed by the ATLAS DAQ team at CERN to manage the fragment indexing and space allocation. This was achieved using a combination of in-memory data structures, block device access and kernel asynchronous I/O. Although such a solution still lacks many features, preliminary results described in [50] show that the Storage Handler object store is capable of sustaining the full bandwidth provided by the storage media.

8.3 Evolution of the ATLAS Storage Handler system

The previous sections illustrated the performance throughput of different solutions for the ATLAS Dataflow system. This was done by combining COTS-based and custom-made technologies. However, this is only one component of the design of a large-scale data acquisition architecture. The required solution for the ATLAS implementation must also satisfy the operational lifetime required for successful data-taking and comply with the overall cost allocated for the project. The following sections will describe these aspects of the ATLAS data acquisition system.

The design of the ATLAS data acquisition system as described in the Technical Design Report (2017) [20] is based on requirements, candidate technologies and cost projections that may not be valid anymore. In fact, after more than four years of market evaluation, prototyping and technology tracking some of the assumptions made in the design are not realistic. In particular, the design of the large storage buffer for the Dataflow system, with the more recent market trends, will be costly to put in place. Figure 8.13 illustrates the historical evolution of the price (in US dollars per GB) of computer storage and memory devices for the past 22 years. The figure includes Hard Disk (HDD) devices, enterprise-grade DRAM and SSD devices as well as 3D XPoint™ devices in both memory format (persistent memory modules or PMEM) and SSD (Intel® Optane™ SSD). In particular, at the time of writing, the storage price per GB of SSDs, which are the main target technology for the ATLAS Storage Handler sub-system, has not followed the decreasing trend assumed in 2017. On the contrary, the current price per GB of SSDs (July, 2022) has increased by over 61% compared to the cost as of 2017. This makes it very costly to build a storage buffer with an expected 1800 SSD devices (see section 6.6). Other storage media (e.g. 3D XPoint™ and HDD) have also followed the same trend [6]. Since the cost projections estimated at the time of the Technical Design Report did not evolve as predicted, the Storage Handler system of the ATLAS Dataflow needs to be re-evaluated.

[6] It is also worth mentioning that, typically, the semiconductor industry follows a cyclical trend in the short run.

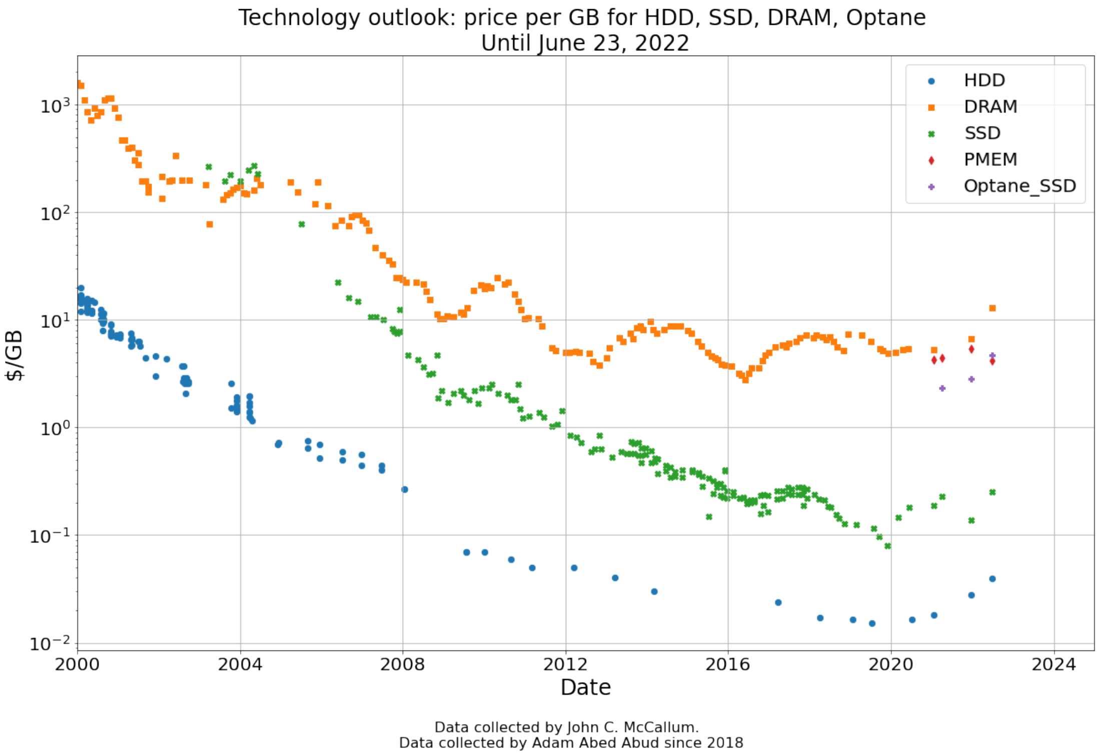


Figure 8.13: Historical price evolution of computer storage and memory devices.

8.3.1 Disk lifetime

One common issue that is experienced when using storage technologies is that such devices can only sustain a maximum number of write operations before experiencing failures. This phenomenon is called SSD *endurance* and it is defined as the total amount of data that an SSD is guaranteed to be able to write under warranty. Two quantities for SSD endurance are commonly used: DWPD and PBW. The former is the *Disk Write Per Day* and it represents how many times per day a drive (given the lifespan of the drive) can be written entirely before experiencing a failure. The latter is the total amount of data in PB that can be written on the drive before a failure occurs. The relation between PBW and DWPD is given by the following:

$$\text{PBW} = (\text{DWPD} * 365)(\text{Warranty [years]})(\text{Capacity [PB]}) \tag{8.1}$$

When designing a storage system it is crucial to take into account the SSD endurance. Otherwise, the risk is that drives will need to be replaced more frequently to obtain the same performance.

Typical enterprise NAND-based SSDs have an endurance that is between 1 and 3 DWPD. As-

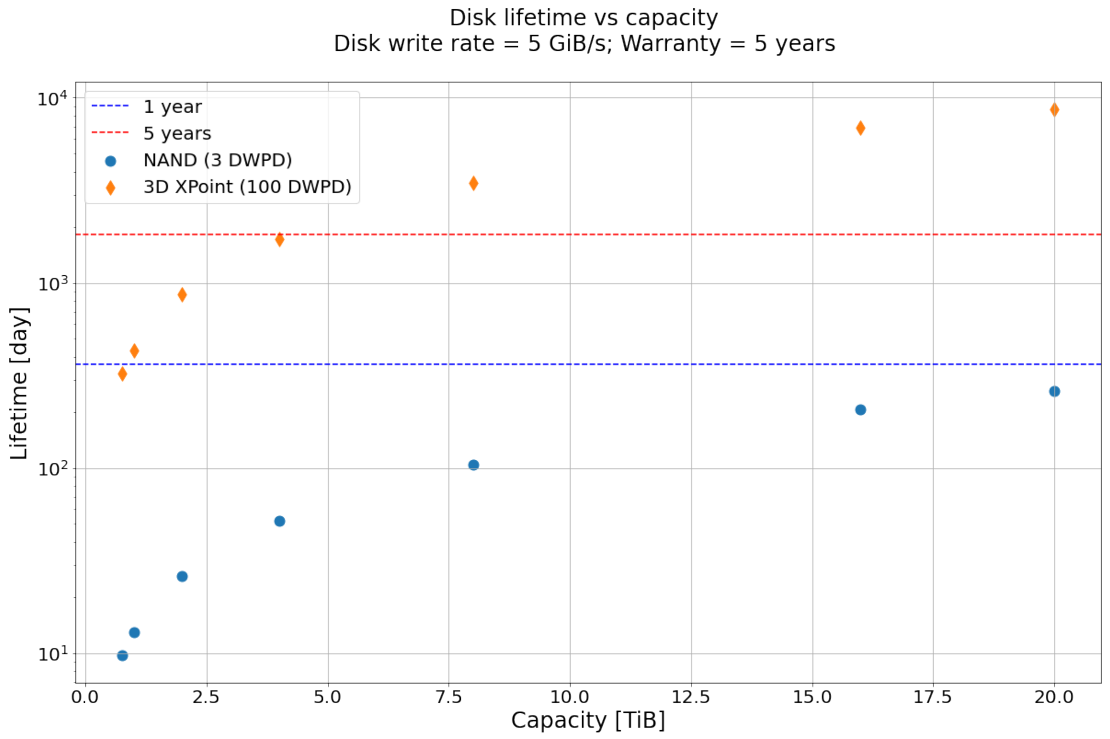


Figure 8.14: Comparison of the lifetime between a NAND-based and an Optane-based SSD assuming ATLAS workloads.

suming a 5 year warranty and a 20 TB drive, this corresponds to approximately 110 PB that a disk can sustain before experiencing a failure. With the write workload of 5 GB/s, the SSD can sustain approximately 250 days which is too low to support the data-taking needs of the ATLAS experiment. On the other hand, the previously introduced 3D XPoint™ technology (e.g. Intel® Optane™ SSDs) provides on average 30 times higher endurance factors as compared to NAND-based SSDs. Figure 8.14 shows the comparison of the lifetime of a NAND-based SSD and an Intel® Optane™ SSD as a function of the capacity of the drives. Assuming a continuous 5 GiB/s disk write rate (ATLAS workload) and a 5 years warranty, even a high-capacity NAND drive has a lifetime that is less than one year. A line indicating the one year and five year thresholds are also shown in the figure. On the other hand, a 3D XPoint™ SSD is capable of continuously sustaining data writes for many years: drives with a capacity of 4 TB can sustain the target write rate for approximately five years of data-taking (ATLAS endurance requirement). Although storage media currently provide the target throughput (e.g. PCIe Gen4 NVMe SSDs with more than 5 GiB/s in write bandwidth) and capacity (more than 15 TB) required to build the ATLAS Storage Handler system, the endurance of SSD devices has not increased. This is another reason why building a robust and safe storage system with the current technologies is unfeasible.

8.3.2 Alternative strategy without a persistent storage solution

The idea why a large persistent storage buffer was proposed for the Phase-II upgrade of the ATLAS data acquisition system was to decouple the L0 triggered data in the Readout System from the processing of the Event Filter. The benefit of such a solution is to allow more flexibility during data taking and to be able to process events when p-p collisions are not occurring during the accelerator inter-fill time. In turn, this makes the overall system more resilient in case of failures like network saturation or misconfiguration of the Event Filter farm. However, as outlined in the previous sections the feasibility of such a storage system is hindered by the unfavourable cost evolution of storage media.

An alternative approach is to build the Storage Handler system using a mixture of different storage media to satisfy the requirements of the system. In this way, it is possible to tune the trade off between lifespan of the devices, maximum buffer size and total cost whilst providing the target throughput needed for the Dataflow system. As an example, using Intel® Optane™ SSDs it is possible to sustain the target rates for many years because they provide enough write throughput and disk endurance. However, as outlined in figure 8.13, the cost per GB of storage of such devices is ten times higher than the one for NAND-based SSDs. An alternative approach would be to use either a combination of NAND-based SSDs with HDD or NAND-based SSDs with Intel® Optane™ SSDs. In the first case (NAND+HDD), the overall Dataflow system would be limited by the lifespan of the devices. In the second case (NAND+OPTANE), it has been estimated that the total data buffer length reduces to roughly ten minutes with a factor of three increase in the total cost of the system.

Overall, building a Storage Handler system with devices that are capable of sustaining the target rates for years of data-taking at the cost projections made in 2017 was considered as high-risk for the data acquisition system. As consequence, buffering the fragment data in a persistent storage media was removed from the baseline design of the Dataflow system and a new architecture for the Phase-II system was investigated.

8.3.3 Simulation studies for the ATLAS Dataflow system

The new baseline design will have an architecture which resembles the ATLAS Run 2 and Run 3 systems [51] where Event Filter nodes will request fragments from the Data Handler nodes and build the full event in their physical memory. Although storage prices did not evolve as expected, networking hardware evolved faster than anticipated. This opens up a new opportu-

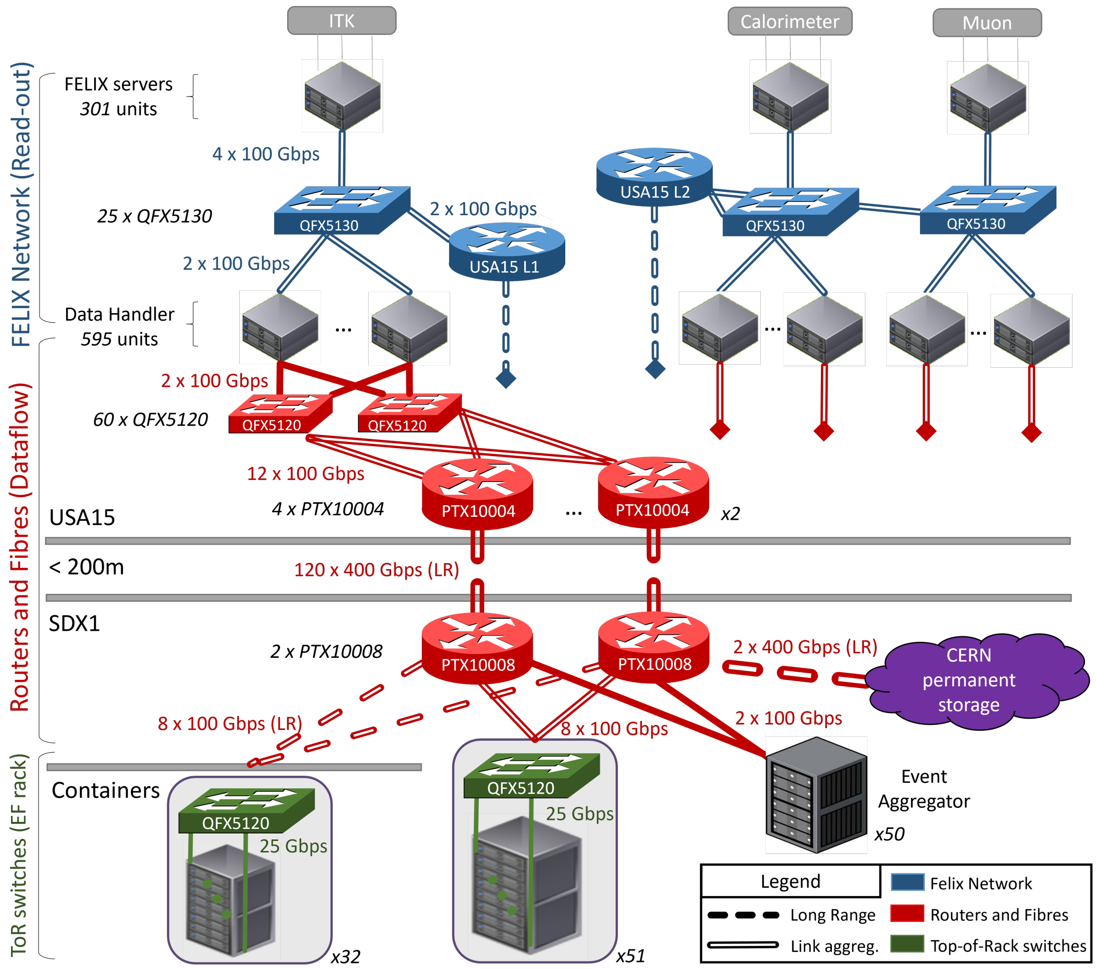


Figure 8.15: Layout of the ATLAS TDAQ network in the evolved design. Figure taken from [52].

nity for the Dataflow system: a more complex network makes it possible to connect all the Data Handler nodes to the Event Filter nodes. Therefore, Data Handler nodes will be able to serve the Event Filter farm at a rate of 1 MHz with a core network capable of sustaining 5 TB/s of throughput. The layout of the updated TDAQ network diagram is shown in figure 8.15. Contrary to the original architecture design illustrated in figure 6.2, the new baseline design does not have a sliced architecture. In addition, the new design also foresees twice the network throughput, making the system now capable of sustaining the full transfer of events to the Event Filter farm.

With the updated network architecture, events are not built incrementally (partial event building) as requested by the EF processing units. Instead, all the fragments relative to an event are sent to an EF node for processing (full event building). This allows performing more complex triggering algorithms as processing units have access to all the fragments of an event. Although this new dataflow method does not provide storage capacity to keep the data, it allows the decoupling between the DH and EF systems. In fact, in the full event building scenario, the task of the DH is to always send all the fragments to the EF nodes. In the case of the partial event building, fragment data are kept in the memories of the DH nodes until a specific EF algorithm requests them; thus, it makes the DH-EF systems coupled. In addition, to provide resilience in the Dataflow system in case of EF operational misconfigurations, the new baseline design will include buffering of the full events in the memories of the EF nodes. In this way, it is possible to temporarily store a large number of events as compared to storing them in the DH nodes. This is due to the fact that the TDAQ architecture for Phase-II foresees 300 nodes for the DH system and 3000 nodes for the EF farm.

In order to investigate the advantages and limitations of the full event building approach, it is crucial to evaluate the impact of this new dataflow approach on the Data Handler system. In fact, the readout software running on the Data Handler nodes will not only route the data fragments from the front-end electronics to the Storage Handler system. Data Handler nodes will also need to handle thousands of network connections with the EF nodes and provide additional buffering in their physical memory until an accept/delete response is received from the EF. Additional buffering in the memory of the DH nodes may be a potential issue for adopting the full event building approach and, therefore, further studies are needed to assess its feasibility for the ATLAS data acquisition system.

Discrete Event System Specification (DEVS) [53] [54] simulations have been used with the objective of evaluating the impact of the full event building approach on the Data Handler system. DEVS simulations are used for modelling a discrete system in terms of *models* (atomic and coupled) and *transitions*. An atomic model represents the dynamic behaviour of an element of the system, whereas the coupled model defines the overall structure of the system (a coupled a model is made with different atomic models). Atomic models are defined by a set of variables that fully describe an element of the system at a specific time. When a *simulation event* occurs (e.g. new packets are generated), models can change state. This is known as *transition*. Note that in the DEVS mathematical formalism, events occur at any point in time (time is continuous) and there needs to be a finite number of transitions (in any finite interval of time). Transitions cannot occur between two consecutive events. DEVS simulations are very powerful tools to model the trigger and data acquisition system of an experiment. In this case, the different components of the TDAQ architecture (Data Handler node, Event Filter node, Supervisor node, etc.) represent the models and the *simulation events* that are responsible for the *transitions* are data fragments generated at the readout. For this research, PowerDEVS [55], an open-source software that implements the DEVS formalism, was used to model the ATLAS TDAQ architecture for the Phase-II upgrade. PowerDEVS provides a C++ framework to model a component of the system (also known as atomic model) and a graphical user interface (GUI) that allows to manipulate and interface the various elements of the system. In addition, PowerDEVS also has a python binding to programmatically define the models in case the GUI interface is not sufficient (e.g. define a coupled model with tens of atomic models interconnected together).

A simulation of the TDAQ architecture was performed comprising 30 Data Handler nodes and 300 Event Filter nodes (approximately 10% of the Phase-II size). These have been modeled with a simplified network layout consisting of only one core router as compared to the planned layout illustrated in figure 8.15. Although, the number of nodes used in the simulation does not match with the expected Phase-II size, the objective of the simulation is to compare two dataflow setups (full and partial event building) and evaluate the scaling of the relevant parameters (e.g. memory buffer size in the Data Handler noes). Note that a simulation also allows to investigate the behaviour of the system with devices and hardware that is yet not available. In addition, reducing the number of EF nodes allows to simulate the system for longer times: in the tests the total simulation time is four seconds. With powerDEVS, data fragments are produced from the DH nodes at a discrete times and with a configurable size (default is 10 KiB) and a configurable L0 rate. A supervisor application is in charge of keeping track of the load on the EF farm as well as assigning the fragments to the available EF processing units. In the case of full event building, all the fragments for a single p-p bunch crossing are assigned to the EF processing unit. In the case of partial event building, fragments are sent to the relative EF processing unit as needed by the filtering algorithm[7]. In the full event building approach, data fragments are deleted as soon as they are sent to the EF nodes and does not allow for re-assignments of the fragments. This has implications on the data safety that were evaluated as acceptable by the ATLAS TDAQ team. In contrast, in the partial event building approach the DH nodes buffer fragments until the event is fully processed and fragment re-assignment is allowed in case a PU fails. This is similar to the Run 2 architecture of the ATLAS experiment.

Figure 8.16 shows a simplified diagram of the components used in the simulation model (Data Handler, Event Filter and Supervisor) all interconnected with one core router. Note that the TCP

[7] In the PowerDEVS simulation, the Event Filter algorithm is represented by configurable distributions representing data rejection and event building processing times, and the number of fragments to request.

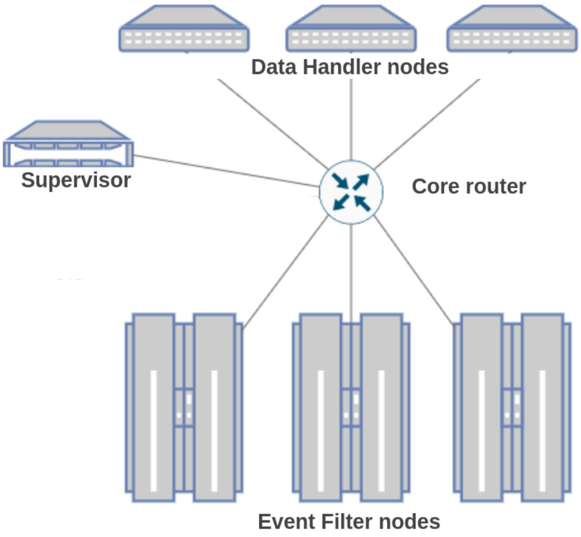


Figure 8.16: Illustration of the architecture used for the PowerDEVS simulation.

network protocol has been disabled in the simulation to avoid any packet drops. Consequently, this allows to achieve a reduced execution time. In addition, since the goal is to evaluate the impact between full and partial event building on the Data Handler nodes, several components of the Phase-II architecture have not been simulated: transfer of the accepted data fragments to the Event Aggregator, timeout, multiple supervisors.

Several simulation trials have been performed as a function of the L0 rate with the goal of recording the utilization of the physical memory of all the Data Handler nodes. This was done for both the partial and full event building approach. Figure 8.17 shows the average of maximum memory buffer size of all the data handler nodes as a function of the L0 rate. As it can be noted, the full event building approach has a much lower impact on the physical memory of the Data Handler nodes as compared to the partial event building. This is also due to the fact that in the full event building approach data fragments are deleted as soon as they are sent to the EF nodes. In the case of 50 kHz the difference between the two event building approaches is almost two orders of magnitude. Therefore, having a network that is capable of supporting the full event transfer of the data makes it possible to adopt an alternative dataflow method for event building. This, in turns, can potentially provide the data fragments for more advanced Event Filter algorithms which can perform a more accurate selection of the data.

As seen from the preliminary results, the new baseline design of the ATLAS Phase-II system will adopt a full event building approach. However, more studies are needed to validate the details of the new baseline design which is still an active research that is being investigated by

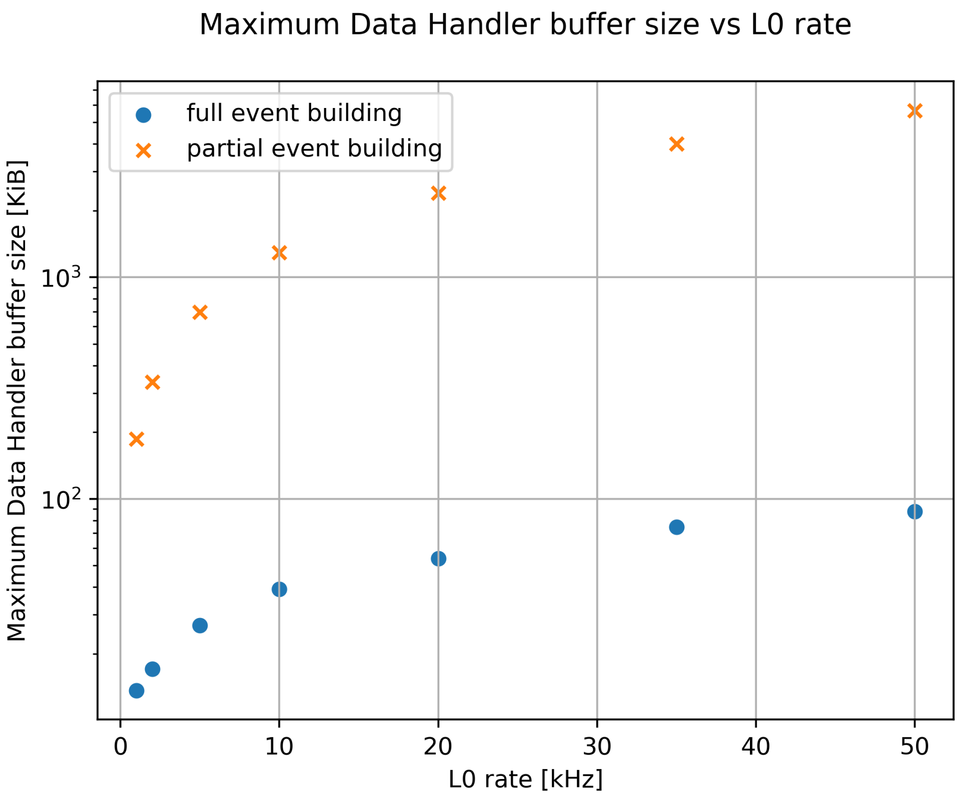


Figure 8.17: Comparison between the full and partial event building approach using a discrete event simulation of the ATLAS TDAQ system. Maximum memory buffer size as a function of the L0 rate.

the ATLAS DAQ team at CERN. This is being conducted through simulation studies, hardware testing and software prototyping.

8.4 Conclusion

This chapter has shown different storage and dataflow methods applied to the data acquisition system of the ATLAS experiment. At first, a custom-made key-value store for generic data acquisition systems (DAQDB) was tested in detail to assess whether the solution can be a viable option for the challenging data acquisition system of the ATLAS experiment. DAQDB was also integrated and benchmarked in detail with the ATLAS TDAQ software framework.

The work shifted towards building an ad-hoc solution for the data acquisition system. For this, two dataflow methods have been tested together: global indexing solutions and local storage management. For the latter, different storage technologies (RocksDB and the ext4 file system) have been tested and optimized in order to achieve the highest bandwidths from the storage media.

The work then focused on the system evaluation of the Storage Handler. Different scenarios have been tested in order to build a cost-effective storage system that is capable of sustaining the target rates and that has sufficient storage endurance. For this evaluation, many factors have to be take into account: size of the buffer, throughput, endurance, overall cost. However, a solution with a storage system was not found to be feasible without a high-risk associated to it. Therefore, buffering data in persistent storage buffer was removed from the baseline design.

Finally, discrete event simulations have been used to investigate different dataflow methods (partial event building vs full event building) in order to validate a new baseline design for the TDAQ system. Discrete event simulations are a powerful tool used to investigate the behaviour of a system at scale. They can also be used to simulate the behaviour of hardware that is not yet available or study in a simplified environment the advantages and limitations of one or more components of the TDAQ architecture of an experiment. It was shown that the full event building approach is feasible and leads to less memory utilization in the DH nodes. Further tests are being performed by the ATLAS DAQ team at CERN to explore different aspects of the new baseline design. Based on the experience gained and the testing performed within the ATLAS data acquisition system, Future directions of this work may also lead to the adoption of discrete event simulations within the data acquisition system of the DUNE experiment.

Contribution of the author:

- Deployment, testing and benchmarking of a custom-made generic key-value store for data acquisition systems (DAQDB)
- Integration of storage technologies within the ATLAS trigger and DAQ framework
- Contribution to the development of the ATLAS Dataflow system: development and integration of several software components
- Evaluation and optimization of local storage management solutions (e.g. RocksDB)
- Studies on the evolution of the storage system for the Phase-II upgrade of the ATLAS experiment (requirements assessment and validation)
- Investigation of the Dataflow system through discrete event simulation studies

9

Conclusion and future work

The data acquisition system is a mission-critical component of particle physics experiments. Its primary role is to buffer, format, select and store the signals from the detectors. The planned DUNE Far Detector module and the Phase-II upgrade of the ATLAS detector at CERN are two experiments that foresee many challenges for their data acquisition system. Both experiments will need to sustain a data throughput of terabytes per second from the front-end electronics. In the DAQ chain, data are then transferred to the Dataflow system in which they are aggregated and served to a computing farm that selects the most interesting signals.

One of the crucial components of the Dataflow system is the persistent storage buffer of the experiment, whose objective is to store the data while a computing farm decides which data to keep for long-term storage. The DAQ of the DUNE-FD module and the Phase-II upgrade of the ATLAS detector are both designed with a large high-throughput storage buffer. Although the use of such a buffer may differ between the two experiments, both systems have similar requirements. This thesis work has been focused on addressing the design, testing, and integration of several storage devices and high-performance storage solutions according to the needs of both the ATLAS and DUNE experiments.

High-throughput storage devices have been investigated to evaluate if they are capable of fulfilling the requirements for the DUNE local storage buffer. Such a system has been designed for a special use case: storing data from neutrino interactions resulting from core-collapse supernova events. The DUNE local storage buffer must be able to sustain, in case a supernova event is detected, at least 100 s of the data stream at approximately 10 GB/s. Two technologies have been evaluated in detail: Intel® Optane™ Data Center Persistent Memory Modules and Micron® X100 solid-state devices. DCPMMs were first tested with synthetic benchmarks and later integrated into a prototype of the DUNE detector available at CERN (ProtoDUNE-SP). Results show that the devices can sustain a continuous stream of data up to 8.8 GiB/s, which is the target throughput required for the DUNE local storage buffer. On the other hand, the Micron® X100 also showed write throughput results close to the target range needed for the DUNE local storage buffer. This performance evaluation was done in a synthetic environment and with the DUNE-DAQ software. In addition, as shown in the preliminary tests, the new generation of PCIe Gen 4 SSD devices used in a RAID 0 configuration also provide bandwidths that are sufficient for the DUNE local storage buffer. In conclusion, both DCPMMs and novel SSD devices show that modern storage media can satisfy the target throughput needed for the DUNE local storage buffer. This makes it possible to focus on a commercial off-the-shelf solution like commonly available storage devices rather than investing effort in developing a custom-made solution.

To further improve the results obtained from the evaluation of high-throughput storage devices, dataflow methods have been investigated to orchestrate the flow of data effectively in a high-throughput environment like the one expected in ATLAS or the DUNE experiments. This research work has investigated the use of COTS and custom solutions for high-throughput data acquisition systems. In particular, this work was done in the context of the ATLAS experiment, where a large storage buffer is designed to decouple data production (Readout) and data processing (Event Filter). At first, a custom-made distributed data acquisition database (DAQDB) was tested and integrated with the ATLAS DAQ framework. Although the write throughput results for a single node were satisfactory, DAQDB was missing many features that made the solution unusable for the data acquisition. Therefore, the effort shifted to a different dataflow method that combines both COTS products and local storage management technologies (RocksDB key-value store). Although the solution was tuned for the dataflow task, it proved to be ineffective in achieving the highest bandwidth from the underlying storage media.

When designing a storage system for high-throughput applications, the size of the buffer and the total I/O bandwidth are not the only elements to consider. The endurance of the storage media, i.e. the capacity of the drives to continuously sustain data writes, is also a critical characteristic to be addressed. At the expected ATLAS write throughput, a storage system made of enterprise NAND-based SSDs would need to be replaced on average after 250 days. This makes it impractical to use enterprise NAND-based SSDs; therefore, modern high-endurance devices (e.g. 3D XPoint devices) need to be considered. However, the unfavorable cost evolution of storage media in the past five years has hindered such an option and challenged the feasibility of having a storage system. Therefore, simulation studies have been put in place to understand a possible evolution of the ATLAS data acquisition system and its implications for the overall dataflow architecture. Based on the preliminary simulations, the ATLAS Phase-II dataflow system has been adapted with a new baseline design that does not include a large persistent storage system.

Finally, the DUNE detector represents the culmination of more than 40 years of research and development in LArTPC detector technology. As a means to further understand the detector, an innovative algorithm (SparseNet) for classifying track and shower energy deposits across the detector was extensively tested. This algorithm has been specifically designed for the classification task using modern analysis techniques based on Deep Learning. At first, an investigation of the relevant physical quantities for the classification task was performed. As a result, nine relevant features were computed from the properties of the hits across the detector. As a second step, the SparseNet classification algorithm was trained and then tested using both simulated Monte Carlo data as well as data collected during the first run of the ProtoDUNE-SP detector at CERN. This was done using a large dataset sample consisting of millions of entries. Results for the classification task show that the SparseNet is capable of achieving an accuracy of over 90% in identifying track and shower hits, outperforming the currently used algorithm in ProtoDUNE. In addition, the purity and efficiency results for SparseNet is more than 90% for both the track and shower classes. This showcases the good discrimination capabilities of the newly introduced algorithm. Therefore, the novel SparseNet algorithm represents a promising solution to be further investigated for the larger DUNE Far Detector modules.

9.1 Future work

As a result of this thesis work, there are several lines of future work that can be further investigated:

- Evaluation of different storage device technologies for the DUNE local storage buffer (e.g. RAID configuration of PCIe Gen 4 SSDs).

- Extraction of supernova events from the DUNE local storage buffer: impact on the Dataflow system.
- Simulation studies to investigate the network impact on the Dataflow system of the new baseline design of the ATLAS Phase-II upgrade.
- Modelling and validation on small-scale setup of different dataflow scenarios to further investigate the ATLAS new baseline design.
- Integration of the SparseNet algorithm into a ProtoDUNE analysis to evaluate the end-to-end performance.

9.2 Dissemination

The results of this doctoral research work have been been published and presented in scientific events in both written and oral format.

9.2.1 Papers

Signing co-author of both the ATLAS and DUNE collaborations. The following is a list of published papers where the author has made significant contributions.

- **Sparse Convolutional Neural Networks for particle classification in ProtoDUNE-SP events**, Proceedings of ACAT 2021 conference, Daejeon, South Korea (main author).
- **Design of a Resilient, High-Throughput, Persistent Storage System for the ATLAS Phase-II DAQ System**, EPJ Web Conf., 2021 (significant contributor).
- **Evaluation of a high-performance storage buffer with 3D XPoint devices for the DUNE data acquisition system**, EPJ Web Conf., 2021 (main author).
- **Experience and Performance of Persistent Memory for the DUNE Data Acquisition System**, Transactions on Nuclear Science, IEEE, 2020 (main author)
- **Let's get our hands dirty: a comprehensive evaluation of DAQDB, key-value store for petascale hot storage**, EPJ Web Conf., 2020 (significant contributor).

The following list of papers are part of further research activities not directly connected with the doctoral work. The author has contributed to the project with the software development of a web-based GUI for control and monitoring of locally-made device for medical applications.

- **The HEV Ventilator Proposal**, arXiv, April 2020.
- **The HEV Ventilator: at the interface between particle physics and biomedical engineering**, Royal Society Open Science, March 2022.

Oral presentations

The following list shows the most relevant oral contributions on the storage and dataflow system of both the DUNE and ATLAS experiments as well as the work on the track/shower classification algorithm:

- **Phase-II Dataflow - Simulation results**. ATLAS TDAQ Open Meeting. June 16, 2022 (CERN).
- **DUNE Data Acquisition system**. Horizontal Drift Far Detector information meeting. December 16, 2021 (CERN).
- **SparseNet for track and shower classification in ProtoDUNE events**. ProtoDUNE Data Reconstruction and Analysis Working group. November 12, 2021 (CERN).
- **Evaluation of a high-performance storage buffer with 3D XPoint devices for the DUNE data acquisition system**. 25th International Conference on Computing in High Energy & Nuclear Physics. May 18, 2021 (Online).
- **Software driven implementation of the latency buffer and supernova buffer with COTS solutions**. Upstream DAQ Readout Technology Review. July 30, 2020 (CERN).
- **Storage systems at the LHC**. DUNE Dataflow meeting. February 26, 2020 (CERN).
- **DAQ Storage technologies** . DUNE DAQ meeting. December 2, 2019 (CERN).
- **ATLAS Phase-II Event building strategy**. ATLAS DAQ meeting. September 3, 2019 (CERN).

Poster contributions

- **Sparse Convolutional Neural Networks for particle classification in ProtoDUNE events**. ACAT 2021. December 1, 2021 (Online).

- **R&D Studies on a Persistent Storage Buffer for the ATLAS Phase-II DAQ System**. TIPP 2021. May 26, 2021 (Online).

- **Experience and Performance of Persistent Memory for the DUNE data acquisition system**. 22nd IEEE Real Time Conference (Second price winner). October 20, 2020 (CERN).

- **Experience with DAQDB: Key-value store for the ATLAS DAQ system**. **Adam Abed Abud**, Danilo Cicalese, Grzegorz Jereczek, Fabrice Le Goff, Giovanna Lehmann Miotto, Jeremy Love, Maciej Maciejewski, Remigius K Mommsen, Jakub Radtke, Jakub Schmiegel, Malgorzata Szychowska. ACES 2020 - Seventh Common ATLAS CMS Electronics Workshop for LHC Upgrades. May 28, 2020 (CERN).

- **DAQDB: key-value store for petascale hot storage**. CERN openlab Technical Workshop, January 22-23, 2020 (CERN).

References

[1] The ATLAS Collaboration, Aad, G et al. "The ATLAS Experiment at the CERN Large Hadron Collider". In: *Journal of Instrumentation* 3.08 (Aug. 2008), S08003–S08003. DOI: 10.1088/1748-0221/3/08/s08003. URL: https://doi.org/10.1088/1748-0221/3/08/s08003.

[2] Abi, B. et al. "Volume IV. The DUNE far detector single-phase technology". In: *Journal of Instrumentation* 15.08 (Aug. 2020), T08010–T08010. DOI: 10.1088/1748-0221/15/08/t08010. URL: https://doi.org/10.1088/1748-0221/15/08/t08010.

[3] Abi, B. et al. "Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume II DUNE Physics". In: *arXiv* (Feb. 2020). arXiv: 2002.03005 [hep-ex].

[4] Berns, L. "Recent Results from T2K". In: *55th Rencontres de Moriond on Electroweak Interactions and Unified Theories*. May 2021. arXiv: 2105.06732 [hep-ex].

[5] Tanabashi, M. et al. "Review of Particle Physics". In: *Phys. Rev. D* 98.3 (2018), p. 030001. DOI: 10.1103/PhysRevD.98.030001.

[6] Aker, M. et al. "Improved Upper Limit on the Neutrino Mass from a Direct Kinematic Method by KATRIN". In: *Phys. Rev. Lett.* 123.22 (2019), p. 221802. DOI: 10.1103/PhysRevLett.123.221802. arXiv: 1909.06048 [hep-ex].

[7] Martínez-Pinedo, G et al. "Neutrinos and Their Impact on Core-Collapse Supernova Nucleosynthesis". In: *Handbook of Supernovae*. Cham: Springer International Publishing, 2016, pp. 1–37. ISBN: 978-3-319-20794-0. DOI: 10.1007/978-3-319-20794-0_78-1. URL: https://doi.org/10.1007/978-3-319-20794-0_78-1.

[8] Acciarri, R. et al. "Design and construction of the MicroBooNE detector". In: *Journal of Instrumentation* 12.02 (Feb. 2017), P02017–P02017. ISSN: 1748-0221. DOI: 10.1088/

1748-0221/12/02/p02017. URL: http://dx.doi.org/10.1088/1748-0221/12/02/P02017.

[9] Abi, B. et al. *Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume I: Introduction to DUNE*. 2020. arXiv: 2002.02967 [physics.ins-det].

[10] Abed Abud, A. et al. "Design, construction and operation of the ProtoDUNE-SP Liquid Argon TPC". In: *JINST* 17.01 (2022), P01005. DOI: 10.1088/1748-0221/17/01/P01005. arXiv: 2108.01902 [physics.ins-det].

[11] Abi, B. et al. "First results on ProtoDUNE-SP liquid argon time projection chamber performance from a beam test at the CERN Neutrino Platform". In: *JINST* 15.12 (2020), P12004. DOI: 10.1088/1748-0221/15/12/P12004. arXiv: 2007.06722 [physics.ins-det].

[12] J. S. Marshall and M. A. Thomson. "The Pandora software development kit for pattern recognition". In: *The European Physical Journal C* 75.9 (Sept. 2015). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-015-3659-3. URL: http://dx.doi.org/10.1140/epjc/s10052-015-3659-3.

[13] Abi, B. et al. "Neutrino interaction classification with a convolutional neural network in the DUNE far detector". In: *Phys. Rev. D* 102.9 (2020), p. 092003. DOI: 10.1103/PhysRevD.102.092003. arXiv: 2006.15052 [physics.ins-det].

[14] Benjamin Graham and Laurens van der Maaten. "Submanifold Sparse Convolutional Networks". In: *CoRR* abs/1706.01307 (2017). arXiv: 1706.01307. URL: http://arxiv.org/abs/1706.01307.

[15] Christopher Choy, JunYoung Gwak, and Silvio Savarese. "4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 3075–3084.

[16] Abed Abud, A. et al. "Separation of track- and shower-like energy deposits in ProtoDUNE-SP using a convolutional neural network". In: (Mar. 2022). arXiv: 2203.17053.

[17] Abed Abud, A. et al. "Reconstruction of interactions in the ProtoDUNE-SP detector with Pandora". In: (June 2022). arXiv: 2206.14521 [hep-ex].

[18] Lehmann Miotto, G. et al. *Trigger and Data Acquisition (TDAQ) System Specifications*. https://edms.cern.ch/document/2679120/2. Accessed: 30-5-2022.

[19] Borga, A. et al. "FELIX based readout of the Single-Phase ProtoDUNE detector". In: *EPJ Web Conf.* 214 (2019). Ed. by A. Forti et al., p. 01013. DOI: 10.1051/epjconf/201921401013.

[20] *Technical Design Report for the Phase-II Upgrade of the ATLAS TDAQ System*. Tech. rep. Geneva: CERN, Sept. 2017. DOI: 10.17181/CERN.2LBB.4IAL. URL: https://cds.cern.ch/record/2285584.

[21] Adam Abed Abud, Giovanna Lehmann Miotto, and Roland Sipos. "Experience and Performance of Persistent Memory for the DUNE Data Acquisition System". In: *IEEE Transactions on Nuclear Science* 68.8 (2021), pp. 2159–2164. DOI: 10.1109/TNS.2021.3084848.

[22] Abed Abud, A. et al. "Evaluation of a high-performance storage buffer with 3D XPoint devices for the DUNE data acquisition system". In: *EPJ Web Conf.* 251 (2021), p. 04013. DOI: 10.1051/epjconf/202125104013. URL: https://doi.org/10.1051/epjconf/202125104013.

[23] A. Bueno, Ines Gil Botella, and A. Rubbia. "Supernova neutrino detection in a liquid argon TPC". In: (July 2003). arXiv: hep-ph/0307222.

[24] Booth, A. *Triggering on Supernova Burst Neutrinos at DUNE*. Zenodo, June 2018. DOI: 10.5281/zenodo.1300453.

[25] Abi, B. et al. "Supernova neutrino burst detection with the Deep Underground Neutrino Experiment". In: *The European Physical Journal C* 81.5 (May 2021). DOI: 10.1140/epjc/s10052-021-09166-w. URL: https://doi.org/10.1140/epjc/s10052-021-09166-w.

[26] Intel. *Intel Optane DC persistent Memory Data sheet*. URL: https://www.intel.la/content/dam/www/public/us/en/documents/product-briefs/optane-dc-persistent-memory-brief.pdf. (accessed: 22.02.2021).

[27] Chandan Kalita, Gautam Barua, and Priya Sehgal. *DurableFS: A File System for Persistent Memory*. 2018. eprint: arXiv:1811.00757.

[28] V. Pentkovski, S. K. Raman, and J. Keshava. "Implementing Streaming SIMD Extensions on the Pentium III Processor". In: *IEEE Micro* 20.04 (July 2000), pp. 47–57. ISSN: 1937-4143. DOI: 10.1109/40.865866.

[29] Izraelevitz, J. et al. *Basic Performance Measurements of the Intel Optane DC Persistent Memory Module*. 2019. eprint: arXiv:1903.05714.

[30] Intel Corporation. Intel SSD DC P4510. URL: https://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/dc-p4510-series-brief.pdf.

[31] Phison Electronics. *PS5016-E16 Gen4x4 NVMe SSD Controller*. URL: https://www.phison.com/en/technologies-gen4/pcie-gen4-awareness/1149-ps5016-e16. (accessed: 25.02.2021).

[32] *Intel Corporation. pmem.io: PMDK*. URL: http://pmem.io/pmdk/.

[33] *Flexible I/O*. https://github.com/axboe/fio. Accessed: 12/02/2021.

[34] Bhattacharya, S. et al. "Asynchronous I/O Support in Linux 2.5". In: 2010.

[35] Sid Lakhdar Riyane. *On the Impact of Asynchronous I/O on the performance of the Cube re-mapper at High Performance Computing Scale*. Sept. 2017. DOI: 10.13140/RG.2.2.27290.90566.

[36] Verbitski, A. et al. In: *Proceedings of the 2017 ACM International Conference on Management of Data*. SIGMOD '17. Chicago, Illinois, USA: Association for Computing Machinery, 2017, pp. 1041–1052. ISBN: 9781450341974. DOI: 10.1145/3035918.3056101. URL: https://doi.org/10.1145/3035918.3056101.

[37] Dageville, B. et al. "The Snowflake Elastic Data Warehouse". In: *Proceedings of the 2016 International Conference on Management of Data*. SIGMOD '16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 215–226. ISBN: 9781450335317. DOI: 10.1145/2882903.2903741. URL: https://doi.org/10.1145/2882903.2903741.

[38] Cicalese, D. et al. "The design of a distributed key-value store for petascale hot storage in data acquisition systems". In: *EPJ Web of Conferences* 214 (Jan. 2019), p. 01014. DOI: 10.1051/epjconf/201921401014.

[39] Abed Abud, A. et al. "Let's get our hands dirty: a comprehensive evaluation of DAQDB, key-value store for petascale hot storage". In: *EPJ Web Conf.* 245 (2020). Ed. by C. Doglioni et al., p. 10004. DOI: 10.1051/epjconf/202024510004.

[40] V. Leis, Alfons Kemper, and T. Neumann. "The adaptive radix tree: ARTful indexing for main-memory databases". In: *ICDE'13*. IEEE, Apr. 2013, pp. 38–49. ISBN: 978-1-4673-4910-9. DOI: 10.1109/ICDE.2013.6544812.

[41] Tonglin Li et al. "ZHT: A light-weight reliable persistent dynamic scalable zero-hop distributed hash table". In: *IPDPS'13*. 2013, pp. 775–787.

[42] Anuj Kalia, Michael Kaminsky, and David Andersen. "Datacenter RPCs can be General and Fast". In: *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*. Boston, MA: USENIX Association, Feb. 2019, pp. 1–16. ISBN: 978-1-931971-49-2. URL: https://www.usenix.org/conference/nsdi19/presentation/kalia.

[43] Josiah L. Carlson. *Redis in Action*. USA: Manning Publications Co., 2013. ISBN: 1617290858.

[44] Zhen Liang et al. "DAOS: A Scale-Out High Performance Storage Stack for Storage Class Memory". In: *Supercomputing Frontiers*. Ed. by Dhabaleswar K. Panda. Cham: Springer International Publishing, 2020, pp. 40–54. ISBN: 978-3-030-48842-0.

[45] Peter Braam. "The Lustre storage architecture". In: *arXiv preprint arXiv:1903.01955*. 2019.

[46] Weil, S. et al. "CRUSH: Controlled, Scalable, Decentralized Placement of Replicated Data". In: *Proceedings of the 2006 ACM/IEEE Conference on Supercomputing*. Tampa, Florida, 2006, 122–es. ISBN: 0769527000. DOI: 10.1145/1188455.1188582.

[47] Adam Abed Abud, Fabrice Le Goff, and Giuseppe Avolio. "Performance evaluation of distributed file systems for the phase-II upgrade of the ATLAS experiment at CERN". In: *Journal of Physics: Conference Series*. Vol. 1525. 1. IOP Publishing. 2020, p. 012028. DOI: 10.1088/1742-6596/1525/1/012028.

[48] Mathur, A. et al. "The new ext4 filesystem: Current status and future plans". In: *Proceedings of the Linux Symposium* (Jan. 2007).

[49] Y. Jia and F. Chen. "From Flash to 3D XPoint: Performance Bottlenecks and Potentials in RocksDB with Storage Evolution". In: *2020 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. Aug. 2020, pp. 192–201. DOI: 10.1109/ISPASS48437.2020.00034.

[50] Abed Abud, A. et al. "Design of a Resilient, High-Throughput, Persistent Storage System for the ATLAS Phase-II DAQ System". In: *EPJ Web Conf.* 251 (2021), p. 04014. DOI: 10.1051/epjconf/202125104014.

[51] Okumura, Y. et al. "Triggering in ATLAS in Run 2 and Run 3". In: *PoS* EPS-HEP2021 (2022), p. 788. DOI: 10.22323/1.398.0788.

[52] Lehmann Miotto, G. et al. *DAQ updated design report*. https://edms.cern.ch/document/2733052/1. Accessed: 24-06-2022.

[53] Bernard P. Zeigler, Herbert Praehofer, and Tag Gon Kim. "Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems". In: 2000.

[54] Matías Bonaventura, Daniel Foguelman, and Rodrigo Castro. "Discrete Event Modeling and Simulation-Driven Engineering for the ATLAS Data Acquisition Network". In: *Computing in Science and Engineering* 18.3 (2016), pp. 70–83. DOI: 10.1109/MCSE.2016.58.

[55] Federico Bergero and Ernesto Kofman. "PowerDEVS: a tool for hybrid system modeling and real-time simulation". In: *SIMULATION* 87.1-2 (2011), pp. 113–132. DOI:

10.1177/0037549710368029. eprint: https://doi.org/10.1177/0037549710368029. URL: https://doi.org/10.1177/0037549710368029.