

Evolution of the Data Quality Monitoring and Prompt Processing System in the protoDUNE-SP experiment

Maxim Potekhin^{1,*}, on behalf of the DUNE Collaboration

¹Brookhaven National Laboratory, Upton, NY11973, USA

Abstract. The DUNE Collaboration currently operates an experimental program based at CERN which includes a beam test and an extended cosmic ray run of two large-scale prototypes of the massive Liquid Argon Time Projection Chamber (LArTPC) for the DUNE Far Detector. The volume of data collected by the single-phase prototype (protoDUNE-SP) amounts to ~ 3 PB and the sustained rate of data sent to mass storage is $O(100)$ MB/s. Data Quality Monitoring was implemented by directing a fraction of the data stream to the protoDUNE prompt processing system (p3s) which is optimized for continuous low-latency calculation of the vital detector metrics and various graphics including event displays. It served a crucial role throughout the life cycle of the experiment. We present our experience in leveraging the CERN computing environment and operating the system over an extended period of time, while adapting to evolving requirements and computing platform.

1 Introduction

The goal of the protoDUNE-SP experiment [1] is a detailed study of a large-scale prototype of the massive 10kt single-phase Liquid Argon Time Projection Chambers (LArTPC) to be used in the DUNE experiment [2, 3]. The experiment took test beam data in 2018 and cosmic ray data throughout 2019. The nature of the experiment requires a versatile Data Quality Monitoring (DQM) system capable of performing complex calculations on a few minutes time scale as well as preserving the results (including various types of graphics and tabulated data) in an easily accessible form and over an extended period of time. In order to support this capability a *protoDUNE prompt processing system* (p3s) was designed, deployed and operated over a period of time of ~ 3 years [4]. Its design proved flexible enough to quickly adapt to different hardware platforms and computational payloads.

2 System design

A predefined (and configurable) fraction of the data written to a local buffer by the protoDUNE-SP Data Acquisition (DAQ) system is transferred to a dedicated space in the CERN EOS disk storage [5, 6] by an instance of the “Fermi-FTS” data handling system [7]. Arrival of new data is detected by an automated process polling the corresponding directory in EOS and triggers creation of job records in the p3s database, forming a queue. The p3s brokerage module then matches these job records to available pilot jobs (running asynchronously

*e-mail: potekhin@bnl.gov

on the worker nodes within the designated resource) and provides the pilot with information necessary to actuate the “payload job” (e.g. as via a URL pointing to a script and a set of parameters). Monitoring information about execution of the payloads is presented to the user in the p3s Web UI. Job outputs are automatically catalogued in the database by a dedicated Web service and are made available to the users via a Web interface through a set of selectors and menus. The following is a summary of the protoDUNE-SP prompt processing/DQM system design features [8]:

- A high degree of automation and a data-driven workflow
- Complete decoupling of the DQM and the DAQ systems, which allows for stable and reliable operation of the latter while being able to make frequent updates to the DQM code and adding features and new types of software modules (payloads) whenever necessary
- A simple pilot-based framework inspired by PanDA and Dirac [10, 11] which minimizes the latency of job submission and provides a layer of abstraction of the computing resources; a Web application coordinates the operation of pilot jobs which in turn manage execution of the DQM jobs defined by the user (the payload jobs)
- A template for JSON-based [9] job descriptions
- Separation of the workload management functionality of p3s from indexing and navigation of the content created by the DQM jobs (a separate “content service”), which provides a clean interface and optimal end-user experience
- Industry-standard tools and frameworks: Django [12] applications run with the Apache HTTP server and the PostgreSQL RDBMS back-end
- Self-describing DQM data which allows automatic and consistent generation of Web pages by the content service for each type of DQM applications without changes to the server code. This is achieved by the requirement that DQM jobs produce metadata (a set of JSON files) which completely describes their output dataset and the logic of its navigation

A conceptual diagram of the prompt processing and DQM system components is presented in Figure 1. The prompt processing and DQM components were implemented as two separate Web services, utilizing separate schemas in a common relational database back-end. Both services can be interfaced via command line clients as well as through full-featured Web GUIs. A separate automated agent (the “Pilot Job Generator”) was run to ensure that the required number of pilot jobs running on the CERN batch facility was available for deployment of DQM payloads. EOS storage was utilized to handle both input data and the DQM data products.

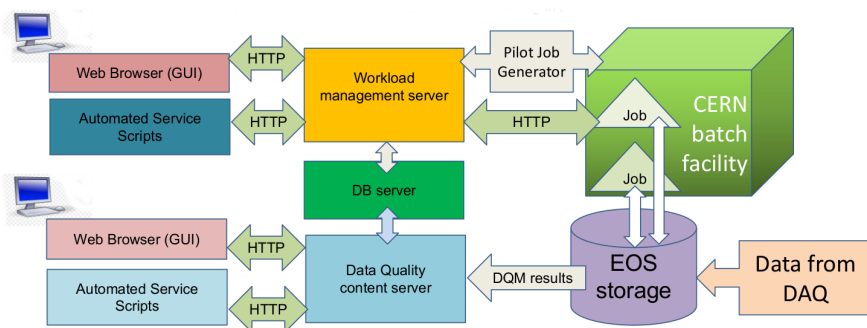


Figure 1. The protoDUNE-SP prompt processing and DQM systems

3 System evolution

3.1 The hardware platform

The p3s pilot-based framework provides substantial flexibility in accessing and managing the available computing resource. Generation of pilot jobs is external to the core system itself and thus can be easily adapted to a particular resource. One basic requirement is that the pilot jobs running on the worker nodes have HTTP access to the p3s server. The first (test) instance of p3s was deployed on an ad hoc collection of local machines. The parallel remote shell (pdsh) was used to manage the population of pilot jobs running on the worker nodes. The next deployment was on a hardware cluster provided by the CERN Neutrino Platform organization [13] consisting of a few dozen worker nodes configured as a HTCondor cluster, and the pilots were managed by scripts utilizing the HTCondor batch system tools. Web and database services were deployed on two nodes which were part of the same cluster. Subsequently a decision was made to leverage the capabilities of the CERN Central Services which included the following:

- Hosting the p3s server and DQM content server on the CERN OpenStack Virtual Machines
- Utilization the central batch facility (“lxbatch”) as the compute element
- Relying on the EOS distributed storage to stage input and output data including serving the DQM content through the Web service

The migration included adapting existing scripts to the new HTCondor environment and using the Kerberized “acrontab” facility to orchestrate their periodic execution. The input data was staged from EOS storage to the worker nodes’ local disk space prior to execution using the *xrdcp* utility.

For the p3s and DQM servers deployed in OpenStack the snapshot feature was used to simplify recovery and to scale the services as needed, for example by transparently increasing the number of cores in a host VM while keeping the software environment intact.

Our experience of long-term operation within the CERN environment is positive. We note exceptional stability of the OpenStack VMs. Intermittent latency fluctuations of file transfers to worker nodes were attributed to periods of extremely high load on the EOS cluster and accounted for. The p3s dashboard was updated with a table of metrics for the batch system, storage space and file system performance to help operators assess the state of the system and take corrective measures.

3.2 The software

Throughout its life cycle the experiment required an evolving set of algorithms to be included in its DQM software, sometimes with frequent software updates. At various points in time the following algorithms were active:

- 2D event display (based on raw data)
- The TPC monitoring application with an extensive set of histograms at various levels of channel aggregation, FFT charts etc (with ~1000 histograms produced in each application run)
- Health monitor for the front end electronics motherboards (e.g. for detection of signal dropouts)
- Prompt reconstruction of cosmic ray track candidates with metrics such as track candidate count and hit count in various parts of the detector

- Liquid Argon purity estimations based on charge values for track candidates; time series were stored in the database and rendered graphically in the Web UI as shown in a screenshot in Figure 2
- Signal-to-noise ratio monitoring (time series stored in the database and rendered graphically in the Web UI)
- Data preparation for the 3D event display implemented on a separate server

System settings allowed to configure the number of jobs allocated to each job type, for optimal resource management. Adding new types of jobs was facilitated by two factors:

- Reliance on existing job templates which required only minor adjustments
- Self-describing DQM output data simplifying integration with the content service [8]

Software updates and its provisioning to the worker nodes were simplified by making use of the CVMFS distribution model.

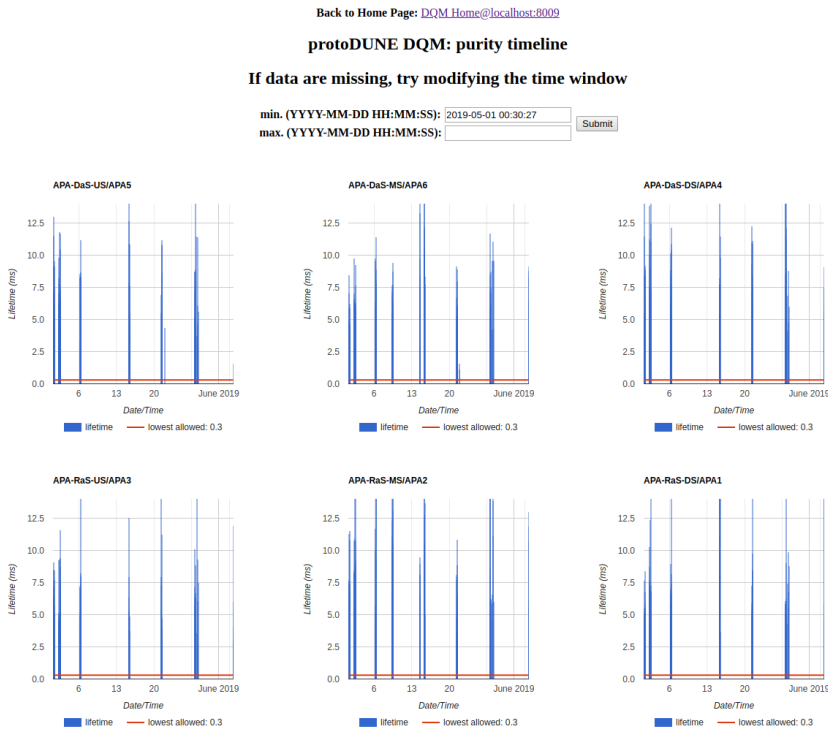


Figure 2. Screenshot of a time series of the electron lifetime in the Liquid Argon of the protoDUNE-SP TPC, based on DQM data for cosmic ray track candidates

3.3 Conclusions

The protoDUNE-SP Data Quality Monitoring system has been one of the crucial components of the experiment. It demonstrated a high degree of flexibility and was quickly adapted to the changing hardware platforms and the evolving DQM software. It is foreseen that the

system will be used in the projected Run 2 of protoDUNE-SP and may potentially serve as a prototype for a DQM system in the DUNE experiment.

References

- [1] M. Spanu, IOP Conf. Series: Journal of Physics: Conf. Series, **1312** (2019) 012003
- [2] R. Acciarri et al., *Long-Baseline Neutrino Facility (LBNF) and Deep Underground Neutrino Experiment (DUNE) Conceptual Design Report Volume 1: The LBNF and DUNE Projects*, e-Print: arXiv:**1601.05471**
- [3] R. Acciarri et al., *Long-Baseline Neutrino Facility (LBNF) and Deep Underground Neutrino Experiment (DUNE) Conceptual Design Report, Volume 4: The DUNE Detectors at LBNF*, e-Print: arXiv:**1601.02984**
- [4] M. Potekhin et al., Proceedings of Science (EPS-HEP2017), **513**
- [5] L. Mascetti et al., J. Phys.: Conf. Series, Vol.**664** (2015)
- [6] S. Fuess et al., J. Phys.: Conf. Series, Vol.**898** (2017)
- [7] A. Norman, *The Fermilab File Transfer System*. e-Print: FNAL CD-DocDB-5412
- [8] M. Potekhin et al., EPJ Web of Conferences **214**, 01026 (2019)
- [9] <https://www.json.org/json-en.html>
- [10] T. Maeno et al., J. Phys.: Conf. Series, Vol.**331**, (2011)
- [11] A. Casajus et al., J. Phys.: Conf. Series, Vol.**219**, (2010)
- [12] N. George, *Mastering Django: Core. The Complete Guide to Django 1.8 LTS* GNW Independent Publishing, ISBN: 099461683X
- [13] F. Pietropaolo, J. Phys.: Conf. Series, Vol.**888** (2017)