EPJ Web of Conferences **214**, 01026 (2019) https://doi.org/10.1051/epjconf/201921401026
CHEP 2018

The Prompt Processing System and Data Quality Monitoring in the protoDUNE-SP Experiment

Maxim Potekhin[1,*], *on behalf of the DUNE Collaboration*

[1]Brookhaven National Laboratory, Upton, NY11973, USA

Abstract.
The DUNE Collaboration is conducting an experimental program (named protoDUNE) which involves a beam test of two large-scale prototypes of the DUNE Far Detector at CERN operating in 2018-2019. The volume of data to be collected by the protoDUNE-SP (the single-phase detector) will amount to a few petabytes and the sustained rate of data sent to mass storage is in the range of a few hundred MB per second. After collection the data is committed to storage at CERN and transmitted to Fermi National Accelerator Laboratory in the US for processing, analysis and long-term preservation. The protoDUNE experiment requires substantial Data Quality Monitoring capabilities in order to ascertain the condition of the detector and its various subsystems. We present the design of the protoDUNE Prompt Processing System, its deployment at CERN and its performance during the data challenges and actual data taking.

1 Introduction

The protoDUNE-SP experiment is designed to study a large-scale prototype of the single-phase version of the Liquid Argon Time Projection Chamber (LArTPC) which will eventually become one of the principal elements of the DUNE apparatus to be constructed at the Sanford Underground Research Facility [1, 2]. The prototype (CERN designation NP04) is located in the CERN North Experimental Hall and is tested utilizing a dedicated beam line from the CERN SPS accelerator complex. The run plan also includes a large number of cosmic ray triggers. Commisioning of the detector took place in September of 2018.

In order to provide the cosmic ray triggering capability a large array of scintillation counters is installed outside of the cryostat. A number of beam instrumentation devices are used to feed the trigger logic and to characterize the beam. Figure 1 shows the view of the top of the protoDUNE-SP apparatus and a part of its readout hardware. There is a dedicated 20 Gb/s network connection from the experiment to the CERN central storage facilities.

The protoDUNE-SP Data Acquisition (DAQ) is equipped with a capable online monitoring system which receives data via the network and does processing in real time. However, certain types of processing, e.g. application of digital filtering to the TPC channels, counting hit candidates *etc.* which belong to the category of Data Quality Monitoring (DQM), require more resources than are available within the DAQ footprint, and include jobs that take substantially longer time than processes run in the online monitor. That could present a resource and data management problem for the DAQ. The online monitor is a real-time system and is

*e-mail: potekhin@bnl.gov

Figure 1. View of the top of the protoDUNE-SP apparatus. The direction of the particle beam is from left to right. Readout systems are visible on top of the detector, with DAQ racks in front.

not well suited for long term preservation and cataloging of the data produced, such as plots, histograms, tables *etc.* as this would substantially complicate the system.

A separate consideration is the necessity to insulate DAQ from potential disruptions due to frequent software updates of the DQM software dictated by the dynamic nature of the test beam experiment. For these reasons a separate *protoDUNE Prompt Processing System* (abbreviated as p3s) was put in place with the following characteristics [3]:

- turnaround time on the scale of minutes (and in some cases tens of minutes)
- lower bandwidth compared to the online monitor (*i.e.* a small fraction of events is processed, but in more detail and with a variety of methods)
- extensive and scalable computing resources provided by the CERN batch facility
- reading data from files committed to CERN central storage, with no direct coupling to the DAQ system
- flexibility in introducing, modifying and configuring software without any risk of disruption of critical DAQ functionality

- substantial storage and browsing capability allowing the users to access and study the DQM results in an efficient manner

2 The protoDUNE-SP Data

2.1 Raw Data Parameters

The detector features the *cold electronics* design in which the amplifiers and digitizers are placed within the cryostat and operate at cryogenic temperatures. There are *warm interface boards* located outside of the cryostat which concentrate data and transmit it to the DAQ systems via an optic fiber. There are 15,360 TPC channels read out at the 2 MHz digitization frequency. There is also a photon detector integrated in the TPC internal structure, but the amount of data produced by this subsystem is substantially smalled than that of the TPC. The size of the data read out from the detector in a single trigger cycle is approximately 230 MB. At the nominal trigger rate and lossless compression applied in the DAQ instantaneous data rate is about 1.5 GB/s, and the sustained data rate to disk is 300 MB/s.

2.2 Data Flow and Distribution

Principal elements of data transmission and storage in protoDUNE are illustrated in Figure 2. Once the raw data are captured by the data acquisition system and written to the online buffer located in the vicinity of the detector in the experimental hall they are picked up by an instance of the *Fermi FTS* data handling system [4, 5] which manages the transfer to EOS [6] (the CERN distributed disk storage system).

EOS serves as the principal staging area [7] from which the data get copied to tape storage at CERN (*CASTOR*) and are also transmitted to FNAL for replication, distribution and offline processing by a separate instance of *Fermi FTS*. Importantly, EOS also serves to stage the input data for the prompt processing system and to store and preserve the various data products produced by DQM jobs. This design ensures a large degree of independence of prompt processing and DQM from the DAQ and vice versa (which is desirable), but it also introduces a latency due to the way data transfer from the Online Buffer operates. This latency is of the order of a few minutes and is considered acceptable in the context of DQM.

3 Data Quality Monitoring and Prompt Processing

3.1 DQM Applications

The following types of applications were developed by the members of the DUNE Collaboration and included in the protoDUNE DQM suite during the detector commissioning:

- monitoring of front-end motherboards
- individual LArTPC channel signal processing (noise reduction, deconvolution)
- basic event visualization in 2D projections
- initial data preparation for an advanced 3D event display running on a separate system
- liquid Argon purity monitoring using cosmic μ tracks
- estimation of the number of hits and charge collected in each section of the LArTPC, RMS of these values
- estimation of signal-to-noise ratio

For a few of these metrics the system produces time-series plots in addition to the tabulated data. The DQM applications are often called *payload jobs* so as to distinguish them from service and infrastructure jobs which exist to support the function of the overall system.

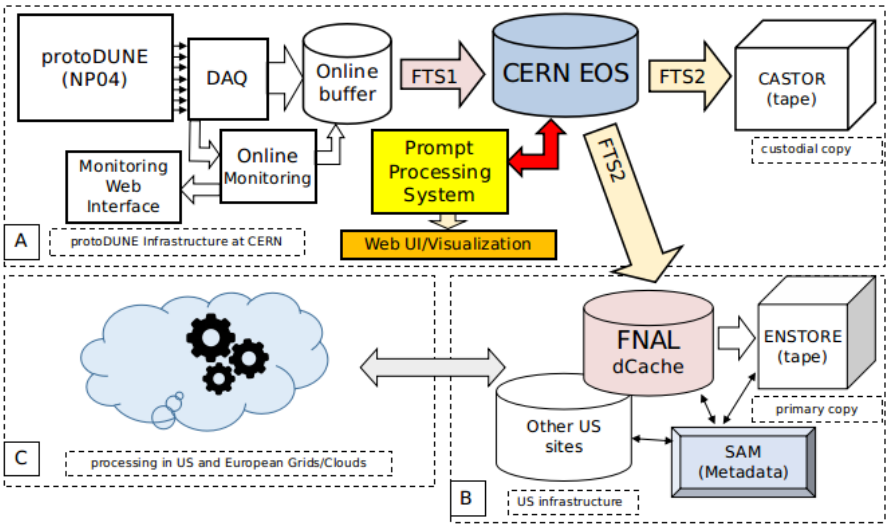

Figure 2. Diagram of the protoDUNE-SP data flow. CERN EOS storage system serves as the principal hub of data collection and distribution.

3.2 Design of the Prompt Processing System

Computing resources for the Data Quality Monitoring in protoDUNE-SP are managed by the *protoDUNE Prompt Processing System* which is separate from both the DAQ and the main production system. This is similar to a few High-Energy Physics experiments which implement *express streams* to perform preliminary calculations on the data very soon after it leaves the DAQ.

The p3s allows the experiment to benefit from combination of high capacity, high performance distributed storage (CERN EOS), CERN networking resources and its Tier-0 computing facility. By utilizing the pilot-based approach [3] to workload management, which is the cornerstone of such well-known systems as PanDA and DIRAC [8, 9], p3s achieves low latency of automated job submission compared to running jobs directly on the batch resource (which is currently HTCondor at CERN Tier-0). In addition, since it combines a web service and a back-end database this approach provides excellent job monitoring capabilities.

3.3 Design of the DQM Content Service

The results produced by the payload jobs managed by p3s can only be useful if there is an efficient interface for the users to access these derived data. Such interface is provided in protoDUNE-SP by a web service accessible through a web browser and a few *CLI* (command line interface) utilities. The DQM output (the "content") can be stored in the system in two ways:

- when the basic unit of data that needs to be stored is a file, e.g. an image in the PNG format, such file is stored on disk (EOS) and its location is recorded in a database so it can be later retrieved and accessed through a dynamically generated web page, based on some selection criteria; this is simplified by the fact that the EOS storage is directly accessible from the Apache server used for this purpose

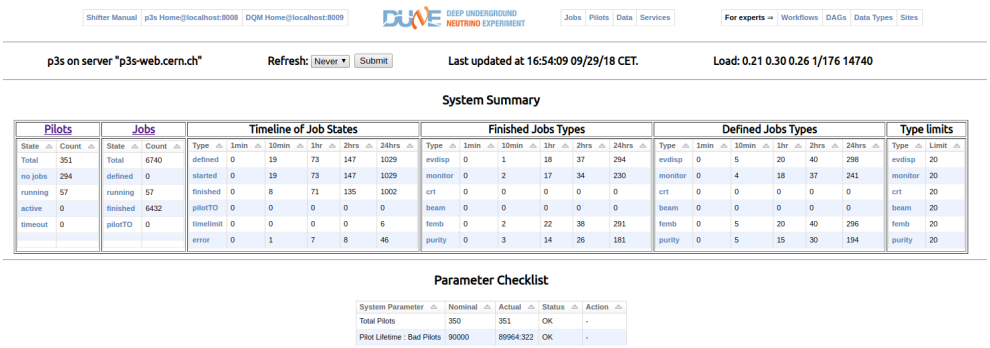

Figure 3. The p3s dashboard which helps to quickly ascertain the state and performance of the system.

- when the basic unit of data is an array of numbers these are for the most part stored directly in the database and are used in dynamically generated web pages either in tabulated format or to generate graphs by using JavaScript

4 Technologies and Interfaces

4.1 Web Services

Both the p3s server and the DQM content server are implemented as Django [10] applications written in Python with standard components such as the Apache web server and PostgreSQL RDBMS as the back-end storage (a few other RDBMS can be used as well). All interactions with either service are conducted via HTTP either from a web browser or one of the included *CLI* utilities. While the p3s instance deployed for protoDUNE-SP is configured to use the resources provided by CERN the system itself is platform-agnostic in the sense that it can manage the resources of any cluster or group of worker nodes and in fact has been succesfully tested in these scenarios as well.

The p3s server is responsible for workload management by matching available pilots to the job requests, and also provides monitoring capabilities by allowing the user to browse and navigate tabulated data describing the state of the various objects in the system (e.g. pilots, jobs, data files *etc.*). The design of the monitoring pages leverages the Django web page template functionality and the well known *django-tables2* package which results in a very small amout of application code. An example of pages served by p3s is presented in Figure 3 (the p3s dashboard).

A suite of CLI clients is provided as a component of p3s for managing pilots, jobs and other entities. In case interaction with the server requires exhange of messages these are formatted as JSON, pimarily because it is trivial to parse it in Python.

There is a centralized log of critical system messages stored in the p3s database which is used to troubleshoot the system. It is also used to automatically generate alarms for the operators should certain parameters fall out of their nominal range.

4.2 The Content Server Interface: the Self-Describing Data

Due to highly dynamic nature of application development in the context of the protoDUNE beam test it is not practical to create static menus, links or make many assumptions about the

Figure 4. An example of dynamically generated DQM menus and links. Each link leads to a page populated with a specific category of plots produced by DQM jobs. Modifying the menus does not require any changes in the server code as it is managed by the metadata produced by the application, which can be changed, added or removed at will by the developers.

categories and content of plots and other data products coming out of the DQM application. Any such information or logic included in the server code in a static manner would result in frequent and time-consuming updates of the server code. Instead, each DQM application is expected to produce two or more files in accordance with simple JSON schemas. One file is referred to as *summary* and it contains basic information about the attributes of the raw data file used as input, such as the run number and a few other indices. This file is stored as a JSON string in the back-end database of the DQM server. It may also optionally contain a number of summary metrics that are then parsed by the page-generating logic of the server and presented in the DQM tables.

The second JSON file (or a group of files) referred to as *description* or *file list* provides references to plots produced in formats such as PNG, as well as description of *categories* and *sub-categories* which are automatically translated by the server into groups of menus with links necessary to access these plots on the dynamically generated web pages. An example of such menus is presented in Figure 4. This approach greatly simplifies integration of different types of payload jobs and data into the DQM content service, with minimal or no changes to the server code.

A few types of data in DQM can be accessed more efficiently if they are presented as time series plots. To handle such cases a web page template was created containing Javascript code making use of Google Charts functionality, so that the time series plots are generated on the fly at the time of the user's request, based on the content of the DQM database.

4.3 Deployment, Testing and Operations

The code of p3s and the DQM content server is version-controlled using *git* and GitHub. The p3s and DQM web services as well as their back-end database server have been deployed on virtual machines of the CERN OpenStack cloud. Processes that need to be run periodically, *i.e.* refreshing the p3s pilot job population, automatic DQM job generation based on detection of fresh raw data written to EOS *etc.* are maintained using *acrontab* which is a kerberized distributed version of crontab used at the *lxplus* interactive Linux facility at CERN.

The system was tested in two data challenges preceding the data taking period, running at data rates and job count exceeding the expected operation parameters by roughly a factor of two. During this testing and actual operation the p3s proved exceptionally stable, with infrequent failures due to outages of some elements of the local infrastructure (e.g. storage and the batch facility) but not the system itself.

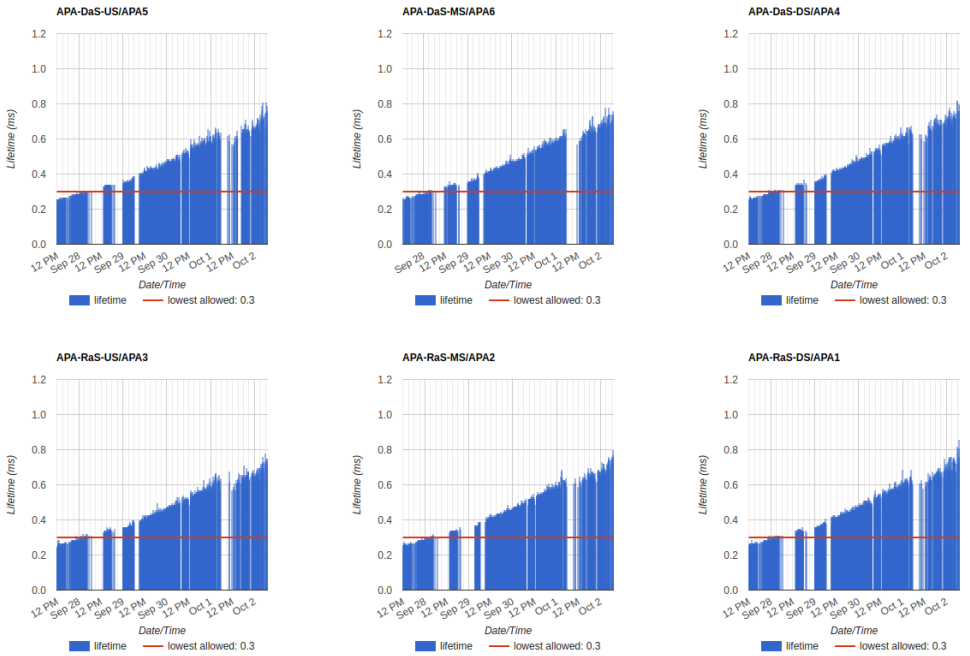

Figure 5. An example of dynamically generated time series plots in DQM. Plots are created using Django templates and Google Charts.

5 Conclusions

The protoDUNE-SP experiment running at CERN in 2018-2019 requires adequate Data Quality Monitoring. To meet these requirements the DUNE Collaboration has developed and deployed web services which manage the execution of the DQM jobs on the CERN facilities and dynamic handling of the DQM data content with efficient user interfaces. The design was based on existing and proven technologies such as Django for the web applicaiton framework, pilot jobs for efficient interface to the batch system, standard JavaScript libraries and a number of others. The DQM software running on the protoDUNE-SP prompt processing system was successfully used during the commissioning phase of the experiment and will remain an imporant part of its ongoing operation.

References

[1] 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**

[2] 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**

[3] M.Potekhin et al. *The protoDUNE-SP experiment and its prompt processing system*. Proceedings of Science (EPS-HEP2017) 513

[4] R. A. Illingworth *A data handling system for modern and future Fermilab experiments. J. Phys.: Conf. Series.* Vol.**513**. IOP Publishing, 2014, doi:10.1088/1742-6596/513/3/032045

[5] A. Norman *The Fermilab File Transfer System*. e-Print: FNAL CD-DocDB-5412

[6] L. Mascetti et al. *Disk storage at CERN. J. Phys.: Conf. Series.* Vol.**664**. IOP Publishing, 2015, doi:10.1088/1742-6596/664/4/042035

[7] S. Fuess et al. *Design of the protoDUNE raw data management infrastructure. J. Phys.: Conf. Series.* Vol.**898**. IOP Publishing, 2017, doi:10.1088/1742-6596/898/6/062036

[8] T. Maeno et al. *Overview of ATLAS PanDA Workload Management. J. Phys.: Conf. Series.* Vol.**331**. IOP Publishing, 2011, doi:10.1088/1742-6596/331/7/072024

[9] A. Casajus et al. *DIRAC Pilot Framework and the DIRAC Workload Management System. J. Phys.: Conf. Series.* Vol.**219**. IOP Publishing, 2010, doi:10.1088/1742-6596/219/6/062049

[10] N. George *Mastering Django: Core. The Complete Guide to Django 1.8 LTS* GNW Independent Publishing, ISBN: 099461683X