Overview of the ProtoDUNE Prompt Processing System (p3s)

Design Paper and Motivation

Supporting documents and an outline of the design can be found in the FNAL DocDB 1861 (authorization required for access).

The p3s is the computing platform for protoDUNE to support Data Quality Management (DQM). Its requirements and mode of operation are different from a typical production system in the following:

- there are stringent ETA requirements for processing jobs since for DQM purposes the results become stale (not actionable) very fast
- only a portion of the data (configurable) needs to be processed
- in any stage of processing a portion of the data unknown apriory can be dropped in order to optimize throughput
- there is no retry mechanism since any substantial delay in processing a unit of data makes the result less relevant (again, the focus is on ETA)
- processing streams are initiated purely automatically and in real time by the data arriving from DAQ
- there is no distinct data handling system for two reasons. First, the cluster which runs p3s is either literally local or can access data through a POSIX-like interface with some modest development and deployment effort. Second, a data handling system would introduce additional complexity, latency and potentially failure modes. Instead, p3s relies on federated storage such as provided by an instance of XRootD. A high-performance NAS could be an alternative. In either case, for purposes of access and processing the data is essentially local on the cluster.

Job dispatch

Pilots

To minimize latency and provide the ability to run transparently on a few local resources (e.g. the cluster at EHN1, CERN Tier-0 and perhaps some other facilities on or around CERN campus) any reliance on the flavor of the underlying batch system needs to be eliminated. In addition, latencies inherent in any batch system should be optimally mitigated. Both problems are addressed by utilizing a pilot-based job dispatch, where the pilots (agents) contact a central service and only receive jobs in case the batch slot is secured and the environment validated. This also combats a few failure modes.

Pilot States

An example of what states a pilot can go through during its lifecycle is given below:

- *active*: registered on the server, no attempt at brokerage yet
- *no jobs*: no jobs matched this pilot
- *dispatched*: got a job and preparing its execution (may still fail)
- *running*: running the payload job
- *finished*: job has completed
- *stopped*: stopped after exhausting all brokerage attempts.

Workflows

Workflow model, interface and the corresponding p3s client are described in a separate document (*WORKFLOW.pdf*).

Location of the input raw data

The protoDUNE DAQ writes the data to its own "online buffer" from which it is transferred to CERN EOS (centralized distributed high-performance disk storage). In order to continue operation in case of a network outage which could make EOS inaccessible, the system mush be able to optionally feed from the online buffer without putting too much extra I/O load on it.

Just-in-time job execution

The near-time nature of prompt processing requires "just-in-time" job submission and not be subjected to the often unpredictable latencies found in batch systems. An efficient and tried way to achieve this is the pilot-based job submission.

Components

- Web service:
- workflows and their templates (dags)
- jobs
- data
- handling of pilots' requests for registration and payload

- Clients
- The *pilot* - submission and management of pilot data on the server
- The *job* - submission of job definitions to the server and management of job data on the server