

Created by: Maxim Potekhin *potekhin@bnl.gov*

Release Notes

- Version 1.01 20180922 first official version
-

Introduction

This document is of interest to members of the DQM and DRA groups who have interest, enough expertise and system access rights to perform mission-critical updates, service restarts and similar actions on the protoDUNE p3s and DQM servers. You will need to obtain credentials for the protoDUNE DQM production account. In this document, we refer to the person performing operations on the servers as *the operator*.

This document is maintained in the “markdown” format and the source file is contained in the directory *p3s/documents/src*. To rebuild the PDF version as needed you can use the command (provided you have pandoc installed):

```
pandoc -s -o EXPERTS.pdf EXPERTS.md
```

Directories

Both p3s and DQM services rely on a shared EOS file system and specifically this folder:

```
/eos/experiment/neutplatform/protodune/np04tier0/p3s
```

Under this directory, there are a few others, of which of most interest are *input* and *monitor*. The former acts as a “dropbox” for incoming raw data files. The latter contains the outputs of the DQM jobs and also is the location from which the DQM Apache server is serving its content i.e. graphs, tables etc.

The output of each p3s job is contained in a single directory in the *monitor* folder which is named according to the job’s UUID. This convention is respected throughout p3s, DQM and various web pages of these systems, which makes it easy to correlate data and access it on disk.

If a directory under *monitor* is removed without deletion of the corresponding DB entry on the DQM server this will result in missing tables, graphs etc for specific runs.

Servers

The machines

There are three VMs running within the CERN OpenStack cloud:

- **p3s-web.cern.ch** - the workload management server which orchestrates the execution of DQM jobs in CERN Tier-0 (“lxbatch”) and has extensive monitoring capabilities via its Web UI.
- **p3s-content.cern.ch** - content management service which allows users to browse and query data products created by the DQM jobs which have run in the p3s framework via its Web UI.
- **p3s-db.cern.ch** - the database server running PostgreSQL which is the critical DB backend for both *p3s-web* and *p3s-content*.

You can access these machines with credentials assigned to the production account *np04dqm*. These can be obtained - on need-only basis - from Nektarios Benekos as authorized by DQM leaders. The user *np04dqm* has *sudo* privileges on these three machines.

The first two servers run Apache-Django services which may need to be updated, stopped, started or restarted.

Maintenance

It’s a good practice to periodically check the available disk space once every few weeks, even though problems are not anticipated in this area. Theoretically, there may be unusually “fat” Apache logs and other leftovers that may need cleaning. Location to be checked is the usual `/var/log/httpd`.

The Linux load report for p3s-web is presented on the landing page of that service, and if the load appears to be too high adjustments may be needed such as reducing the frequency interaction of the p3s pilots with the service. This is usually not necessary if there are only a few hundred pilots running (200-300) but may become an issue at the pilot population of over a thousand. The number of running pilots is readily available on the landing page.

Updates

The database server p3s-db does not need updates as it runs a static installation of PostgreSQL.

The servers **p3s-web.cern.ch** and **p3s-content.cern.ch** may need updates depending on circumstances. If the developer made changes to the service and/or expert shifters opted to change bits of the server configuration and these changes are in the repo, the p3s and DQM servers are updated in the following

way - the operator issues the “git pull” command in the directory “~/projects/p3s” and then the Apache server needs a restart. There is an alias on both machines under the name *arestart* (for “Apache Restart”) which will ask for the user’s password. The operator must be logged on the machine as. *np04dqm*.

If a server tanks (which happens extremely rarely and not really expected during the run) the operator will just need to restart. If the machines are inaccessible this requires immediate escalation via the CERN ticket system at a high priority. The operator may also check the status of these machines in the OpenStack cloud via its Web interface. These are assigned to DUNE.

Crontab

There are a few service processes that allow p3s/DQM to run according to the design. These run periodically as directed by crontab entries defined by the user *np04dqm*.

At CERN there is a Kerberos-enabled distributed version of crontab called **acrontab**. To query the current contents the command *acrontab -l* must be used. A sample output is presented below:

```
01,11,21,31,41,51 * * * * lxplus.cern.ch /afs/cern.ch/user/n/np04dqm/public/p3s/dqmconfig/ht
05,25,45 * * * * lxplus.cern.ch /afs/cern.ch/user/n/np04dqm/public/p3s/dqmconfig/services/p
07,17,27,37,47,57 * * * * lxplus.cern.ch /afs/cern.ch/user/n/np04dqm/public/p3s/dqmconfig/ht
1,31 * * * * lxplus.cern.ch /afs/cern.ch/user/n/np04dqm/public/p3s/dqmconfig/services/tscan.
```

The contents of crontab are version controlled in the DUNE repo under “dqm-config”. The contents can be found at

https://github.com/DUNE/dqmconfig/blob/master/crontab/np04dqm_prod.cron

It is important to keep this under version control, so if you need to modify this, please clone the directory, check in and push the content, then repopulate crontab as the user “np04dqm”. If you need to temporarily stop all crontab activity, this can be achieved by

```
acrontab -r
```

which deletes the crontab content. It can be reinstated by

```
acrontab < /np04dqm_prod.cron.
```

The first three entries perform maintenance and cleanup of the p3s pilot population. The line on the bottom of the list is driving a script which scans the p3s input directory for new arrivals and submits p3s jobs in a few different categories. The script is in a public directory and you can inspect its contents, it is also of course in the git repository:

https://github.com/DUNE/dqmconfig/blob/master/services/tscan_np04dqm.sh

The command line arguments to this script have the following significance:

- -5 is the number of minutes defining the first point in the past when the system is looking for new files. If the script is activated at 12 p.m., it will scan files created between 11:55 a.m. and 12 p.m. If a wider interval is needed, the file np04dqm_prod.cron must be updated and uploaded to crontab.
- np04 is a wildcard i.e. the starting portion of the raw file name. If it changes (unlikely but possible) this needs to be updated
- Z suppresses diagnostic output from this command. If you choose to run it interactively - which is often very useful - you may want to set this argument to “D” which will produce debugging information.

In fact, this command can be used to process data for a specific protoDUNE run, e.g.

```
/afs/cern.ch/user/n/np04dqm/public/p3s/dqmconfig/services/tscan_np04dqm.sh -5 np04_raw_run00
```

where 4610 is the run number. Padding must be respected for this to work. If the run was a while back, just change -5 to a larger negative number so the files are still picked up.