IDUNE Analysis and Production Storage Element Requirements

April 10, 2019

v0.0.0a

Tom Junk, Michael Kirby, Heidi Schellman, Steve Timm

Introduction

A common problem within interactive compute environments is that there are several different access patterns for stored data. DUNE has found that the currently implemented available storage elements have limitations that cannot meet the needs of the experiment without significant development or reconfiguration. In order to help guide the choices to be made about future solutions, this document intends to layout the requirements and workflows impacting the storage elements that are available to interactive computing at Fermilab for DUNE collaborators. The intent is to present this document to SCD, SPPM, and service providers as part of the process of reconfiguration and updating the disk and storage elements.

DUNE Activities

There are five main activities which place requirements on the storage elements available to DUNE collaborators: interactive code development, interactive testing of software, analysis jobs on the grid[1], production operations, and ntuple analysis. Each workflow is essential to the success of the DUNE experiment, from the design and prototyping of the detectors, to the collection of data and the production of published physics results. Additionally, there are requirements from the DUNE collaboration in terms of management of the available resources. The management of resources is included to ensure that collaborators are able to be productive regardless of actions of other collaborators or experiments. We will now describe in detail the workflows and the requirements on storage element that each imposes.

Interactive Code Development

Collaborators install source code and associated files from one or more git repositories into a working area, modify them, and then build new libraries and binaries from the modified source code in order to run new algorithms, simulation, etc. These binaries, libraries, and associated files must be located on disks that allow the files to be modified and programs to be executed.

[1] The "grid" here covers all HTC and HPC computing at scale accessible through HEPCloud, FNAL jobsub service, or any other OSG/WLCG scheduling mechanism.

DUNE reserves a special computer as a “build node” devoted to compiling software and on which other activity is prohibited. Having a storage volume with “bare-metal” performance is important to decrease the development cycle time (edit, compile, build, test, etc) So there is a requirement for fast access (both throughput and IOPS) to a volume where built software can be located. For transition of the built software to non-build nodes, we require access to disk space available on a network that is also available to other interactive machines where software can be tested. As the build node is dedicated exclusively for software compilation and building, allowing software tests is generally prohibited as the large memory requirements of DUNE software would impact other collaborators’ ability to use the build node for its intended purpose. The build node therefore does not have data storage disks mounted (e.g. /pnfs/dune).

Interactive Software Testing

In order to test built software, small samples of data for input and output must also be available on non-build, interactive nodes, though not necessarily on the same storage systems that the code and libraries are located. Usually, special effort is not invested in order to create these small samples -- instead, interactive running of tests using a small fraction of a much larger dataset which may not be accessible on the interactive nodes.   In order to test software properly, the output must be stored somewhere, read in, and checked. Often the output of tests can span an hour or more of clock time and therefore the output volume should be able to handle long delays between file open, modify, and close. If an interactive, test job flushes records infrequently because of lengthy calculations or delays in obtaining input, the output file should not be locked for further updates causing tests to fail. Therefore, it is seen than an output storage with full POSIX capability that can accept writes on arbitrary timescales is required.

The DUNE Computing Consortium needs a mechanism to limit the activities of individual users and analysis groups from saturating or consuming an entire resources without overburdensome amounts of oversight/management. Care must be exercised so that new collaborators unfamiliar with the resources don’t scale up their testing workflow using the interactive testing methods, and overwhelm with the storage elements or interactive node resources. Limiting users ability to write large volumes of data to POSIX nodes is one strategy to accomplish this. We would require the ability to limit users write access to specific volumes and directories, allow users to read from all volumes, and be able to limit both groups and individual usage storage limits. Hopefully, the ability to expand and contract a volume without data migration would be possible (assuming that there is empty storage that allows for the operation). Access to larger storage elements should be either through a POSIX-like listing mechanism, or through a successfully supported file catalog and streaming protocol that allows for quick and efficient location of files.

Analysis Jobs on the Grid

Grid analysis jobs need to read large amounts of data, and generally produce output data files that are orders of magnitude smaller than input data volumes. The output data is usually in the form of ntuples or histograms, and can even take the form of text files or other output. The large input datasets are commonly read from shared storage elements over the network with the a small number of jobs accessing any single data file. Access to these files should be available through both streaming and full file copies. Along with access to data files, analysis jobs will commonly involve user-built, custom software that will need to be distributed to every worker node. These libraries can change frequently (O(10 times) per day) and so are not well suited for distribution through CVMFS or similar cache based library distribution services. A storage volume with high bandwidth and ability to handle O(1000) simultaneous connections of GB or similar size files is required. There is no requirement that this volume be mounted on the interactive nodes, but tools must be available to transfer tarballs there and list content of the volume.

The output files from analysis grid jobs are often small. But there are often many of them, and they often must be combined together somehow after the grid analysis jobs have all finished, and checked while the grid jobs are running. A grid analysis project may produce tens or even hundreds of thousands of files that are of order MB in size or smaller. Some users also run their own Monte Carlo production projects using their Analysis role on the grid. The storage and workflow is similar to Production Operations described below, but usually on a smaller scale than experiment-wide production projects so the total output from an analysis user can still be large O(10 TB).

High capacity storage element needs to be able to handle O(10,000) simultaneous connections, and limit any individual user to O(1000) simultaneous connections. Software will retry if connection fails with exponentially increasing delay between retries. Again, the DUNE Computing Consortium needs a mechanism to limit the activities of individual users and analysis groups from saturating or consuming an entire resource on this high capacity storage volume where grid output will be stored. This is a new requirement from the current solution implemented through DUNE dCache pools.

Production Operations

Production jobs make use of local scratch space on the worker node, and copy outputs back to FNAL dCache areas (and eventually SEs at other sites). Typical jobs stream their input files, also reducing I/O load on the worker node disks. Several thousand simultaneous jobs are commonplace, each of which can be expected to make a streaming connection. Output files tend to be typically hundreds of MB to a few GB, larger than typical analysis output files. Production does need a small amount of interactive storage space for such activities as debugging, creating configuration files, custom tarballs, fetching logs, and running small interactive test jobs. However the total size requirements are much less than those expected for an analysis. Production also makes use of the resilient dCache area for staging tarballs to jobs, so some functionality along those lines is also a requirement (the planned rapid CVMFS area would also meet those requirements).

Ntuple Analysis

The final step in the preparation of physics results is the creation of plots from reduced data produced by user analysis jobs on the grid in the form of ntuples. Typically histograms are filled and plots made, and often calculations must be run on the contents of the ntuples or histograms, which sometimes requires another stage of analysis jobs on the grid. The ntuple analysis step is labor intensive but much less CPU intensive than analysis jobs on the grid, although specialized calculations like Feldman-Cousins confidence region calculations may require large CPU resources with rather small inputs. Proper interpretation of the data may require reading the ntuples many times in order to evaluate the impacts of systematic uncertainty, to identify and fix mistakes, to address comments from reviewers, and to prepare plots and numbers for publication. Given the iterative nature of this step, the speed with which the I/O operations complete is important.

Ntuple analysis on interactive nodes will require a storage element with most POSIX capabilities or sufficient software tools to allow for effective listing, location, and streaming of files. Files need to be mutable, directories accessed through standard Unix commands (e.g. ls, cp, mv, rm, etc), and files have user ownership and access control. Additionally, the ntuple files should be available across multiple interactive nodes in order to allow for analyzers to shift to resources not overloaded (i.e. not everyone running on dunegpvm01.fnal.gov). Again, the DUNE Computing Consortium needs a mechanism to limit the activities of individual users and analysis groups from saturating or consuming an entire resource.

Summarized Requirements

- Production and Analysis code development requires performant POSIX-like capable volume for compilation and build executables.
- Interactive software testing and Production operations requires a POSIX-like capable volume for reading and writing of output of test jobs. This volume should be available from all interactive nodes through some mechanism. The ability to limit read and write access and establish limitation on group and individual usage should be available. Access to larger storage elements should be through a POSIX-like listing capability or using tools of similar capability.
- Analysis grid jobs require a storage volume with high throughput in terms of both bandwidth and number of active connections (O(1000) per user) for the distribution of custom libraries and configs. This volume does not need to be mounted on interactive nodes, but some mechanism to list and remove custom libraries and configs should be available.
- Analysis grid jobs should have access to a high capacity storage element that can handle O(100) connections from a single user and output datasets O(10 TB) in total size.

The total number of connections for the experiment should be O(1000) to cover Production and Analysis workflows and there should be a mechanism to limit the number of connections from a single user with the burden of retry on the client side. A mechanism should be available to limit the total storage volume utilized by individual and group along with control over write permissions per user and group.

- Ntuple analysis requires a storage volume where files are fully mutable along with POSIX-like access through either unix commands or similarly provided and supported tools to accomplish those tasks.