# MicroBooNE Offline and Computing Review February 2015

## Introduction

The MicroBooNE project is almost complete and the experiment is expected to take data in FY15. The experiment spokepersons and SCD management would like the committee to review and evaluate the offline computing software and infrastructure readiness of the experiment for successfully carrying out planned commissioning and physics analysis tasks. In particular, the review should comment on:

1. The current offline computing infrastructure and tools, including build and release tools, simulation tools, framework, database, workflow, workflow management, data management, and operations. Is the experiment efficiently leveraging tools and expertise provided by SCD? Does the experiment have sufficient resources from SCD?
2. Are the tools, infrastructure, and established processes sufficient to engage non-expert resources from the collaboration? Are best practices employed in these processes?
3. The manpower needs and availability, both from the experiment and SCD.

The committee is charged with producing a written report addressing these questions and making recommendations for correcting any problems and issues identified. This Lehman-style report will present findings, comments, and questions to each of the questions in the charge.

## Committee

Adam Lyon, Andrew Norman, Burt Holzman, Jim Amundson, Jim Kowalkowski, Jon Paley, Marc Mengel, Parag Mhashilkar, Peter Shanahan, Tom Junk, Oliver Gutsche (chair)

## Executive Summary

MicroBooNE is close to the start of data taking and presented a comprehensive picture of the preparation of the online and offline software and computing systems. An enormous amount of work has been already done to get ready for data taking, and the collaboration is to be congratulated for this. The collaboration is using many of the centrally provided software packages and tools. SCD is to be congratulated for enabling many experiments, including MicroBooNE, with the same toolkit efficiently and successfully.

The committee has recommendations for the MicroBooNE collaboration and SCD to increase the preparedness of MicroBooNE for the start of data taking and the preparedness for continuing data taking in the next years.
The committee recommends to MicroBooNE to create a resource-loaded integration and commissioning schedule. The schedule needs to list the deadlines for different activities and their interdependencies. It needs to also contain an effort plan to schedule the most scarce resource of MicroBooNE: expert personpower, efficiently and without conflicts.
The committee recommends to MicroBooNE to critically review procedures before the start of data taking, especially in the area of operating the data taking chain and the computing systems, release procedures and procedures that ensure reproducibility and coherence of the final physics results.
The committee recommends to MicroBooNE to critically review their plans for duplication of systems that are already being offered by SCD as part of the FIFE toolkit. We advise MicroBooNE to create maintenance and support plans for systems that have been developed by MicroBooNE.
The committee recommends to SCD to continue working on solutions for the Continuous Integration system (CI) and the auxiliary file access from non-FNAL computing resources. The LArSoft effort is suffering from lack of development and integration personpower, which is not only in the interest of MicroBooNE to be solved.
In the following, the committee presents its detailed findings, comments and recommendations.

# 1. Current offline computing infrastructure and tools

The review should comment on the current offline computing infrastructure and tools, including build and release tools, simulation tools, framework, database, workflow, workflow management, data management, and operations. Is the experiment efficiently leveraging tools and expertise provided by SCD? Does the experiment have sufficient resources from SCD?

### Computing Frameworks

The MicroBooNE experiment uses a combination of three different analysis frameworks which address the data processing and analysis needs of the experiment in different manners. We address our findings for each of these frameworks separately and in the context of what they are intended to address.

### art/LArSoft

The LArSoft software suite is a large set of common tools designed to facilitate large-scale data processing and data analysis across a wide array of liquid argon TPC-based experiments. It is supported by the Fermilab Scientific Computing Division and is based on the art framework which has a significant user base across both in the neutrino community and in other high energy physics communities (e.g. precision muon physics).

Findings

There are over 100 open issues in the Redmine issue tracking system relating to the art and LArSoft software. These issues are not all directly related to MicroBooNE feature or bug reports nor do they represent issues that currently prevent the MicroBooNE experiment from using the software to complete their primary data processing work.

The primary build/release system for the art/LArSoft environment is provided by the MRB system. This system relies on cmake for dependency checking and enforces fully consistent builds through its repository approach to release management. Using the MicroBooNE interactive VMs, the MicroBooNE experiment reports build times of approximately 15 minutes for the portions of their software that are under development.

The MicroBooNE experiment reported that there was a lack of documentation regarding the best practices that should be used when working in the art/LArSoft environment. In particular they requested information regarding art::Services and their usage.

There is currently no interface integrated with the MicroBooNE art/LArSoft environment which provides access to the calibration database. There exists a calibration database interface, but it has not yet been integrated with MicroBooNE's code. There is similarly no interface to online DAQ databases in the offline environment.

Comments

The number of open tickets in the Redmine issue tracking system is indicative of both the level of activity that is occurring within this project and the level of effort that is available to it. A "backlog" of tickets which are unable to be handled by the amount of developer effort allocated to the project is a need for concern, while a high volume of tickets/issues which are being addressed in a timely fashion is an indicator of a very healthy and active project with broad acceptance across its user community. Examination of the Redmine issues for the art project reveals no high priority issues in the queue, no reported bugs in the queue. The remaining tickets are a combination of accepted feature requests (28), maintenance items (17), and feedback items.

The MicroBooNE builds via MRB/cmake and are performed over networked filesystems on virtual machines with low core counts. Other common build systems use different paradigms for their build and release architectures. Most notably, the SoftRelTools (SRT) toolset employs a "base release/test release" architecture which significantly reduces build times for uses who are working on a single module or small segments of a project. However, this system has its own deficiencies and can permit inconsistencies across builds. The cmake-based system was specifically chosen by the art developers to avoid this problem, at the cost of increased build times in some instances if certain best practices are not employed. Build times with MRB for the art/LArSoft software can be significantly reduced through proper configuration and use of best practices. It is also possible that the development VMs are not adequate for the MicroBooNE build tasks.

There is significant documentation regarding the art framework, including a detailed workbook and classes that have been hosted by Fermilab to teach the use of the framework to researchers. This documentation already covers in detail the topics which the MicroBooNE experiment reported were missing. Moreover, other high energy physics experiments, most notably NOvA and Mu2e, have used this documentation successfully to train large numbers of researchers and have developed large code bases which make use of these features.

The lack of interfaces to the different databases that the MicroBooNE collaboration will need to access from their offline software is troubling. Concurrent access to databases from large numbers of client jobs is a scaling problem that has been addressed by prior experiments and for which there are well-documented solutions. There are already application level interfaces in art/LArSoft to similar conditions and calibration databases that are in use by the NOvA and Minerva experiments, including permitting access to other “simple” custom databases schemas. All of these solutions have been engineered to scale to large numbers of concurrent accesses through caching web backends. Some of these interfaces are even implemented as modules and services within the art framework and could be adapted readily to fill the needs of the MicroBooNE experiment.

Recommendations

To the SCD:

The SCD should audit the open tickets for the art framework to identify any open issues which negatively impact or prevent the MicroBooNE experiment’s ability to perform data processing. These issues should be re-prioritized to ensure that development effort is directed appropriately.

The SCD should examine existing documents regarding the art framework and the LArSoft software to identify deficiencies and specifically improve documentation on best practices that should be employed when working with the art/LArSoft suite. The art and LArSoft teams should continue to attend regular MicroBooNE meetings where software developments are discussed. Future SCD-sponsored training sessions should give more focus to art and LArSoft and developing algorithm within these frameworks.

The SCD should provide information on the proper configuration and use of the MRB build/release system to improve the build times of the MicroBooNE code. The SCD should also provide recommendations on an effective architecture (interactive or batch) which the MicroBooNE and other modern experiments can use for algorithm development and frequent build/rebuild cycles that typify code development. In particular, the SCD should examine the current 4-core virtual machine (VM) development nodes with network attached storage that are in use by the experiment, and determine if the observed build times can be improved through reconfiguration of the nodes or their storage. Based on these investigations, the SCD should determine if policy and resources needs to be adjusted to match the needs of MicroBooNE’s development practices.

The SCD should investigate common solutions for database access in art/LArSoft.

To the Experiment:

The MicroBooNE experiment should examine the extensive documentation that has been developed for the art framework and should avail themselves of the training opportunities that are offered regarding the art framework and modern scientific computing. The community support email lists should be used more frequently for questions regarding best practices and use of facilities. The leadership within the MicroBooNE collaboration should be aware of the documentation and training materials and should engage in efforts to disseminate the existence of this information to their researchers.

The MicroBooNE experiment should develop offline interfaces to their conditions/calibration data, DAQ/run data and other databases which build on the experience and models already in large scale production by other experiments. The MicroBooNE experiment should identify effort within the collaboration which is capable of working with SCD framework and database experts and experts on other experiments to develop these interfaces prior to the start of commissioning.

## LArLite

The LArLite framework has been designed by a post-doctoral researcher within the MicroBooNE experiment. It is designed to provide a framework for data processing and data analysis and algorithm development that is lighter weight than the framework provided by art/LArSoft. The LArLite framework has a broad user community within the MicroBooNE experiment.

### Findings

The MicroBooNE experiment presented that LArLite was developed due to specific deficiencies in the art/LArSoft suite. In particular, the experiment noted the performance of a specific algorithm, the build times for code in the art/LArSoft environment, the learning curve necessary to do use the full framework, and the configuration of modules and jobs bringing in additional configurations for dependent modules.

The MicroBooNE experiment currently develops, maintains, and deploys algorithms and data products in both the LArLite and art/LArSoft environments. Maintaining this dual deployment scheme requires porting of algorithms and data products between the two frameworks in both directions (LArSoft to LArLite and LArLite to LArSoft). This porting is currently non-trivial and is performed primarily by the maintainer of the LArLite framework.

The need for improved algorithmic structure, better division of labor, and performance has led to a redesign of the MicroBooNE and LArSoft data products. The effort involved in porting code in both directions has led to changes in the rubrics for how to develop algorithm code. The experiment requested the implementation of framework-independent algorithms.

### Comments

The deficiencies that were cited by the MicroBooNE experiment as the impetus for the development of the LArLite framework need to be reexamined. Some appear to have definite solutions, which have already been implemented (as is the case in the performance of a specific algorithm). Some need to be applied to the MicroBooNE releases, such as better use of the MRB build system to reduce build times. Some are unrelated to the actual framework and are a reflection of working within large and complex software suites.

In particular, the issue that MicroBooNE cited regarding the configuration of art modules bringing in additional configuration of dependent modules is not a deficiency of the art framework, but a boon to it. Modern software projects on the level of MicroBooNE, NOvA or other large modern neutrino experiments naturally involve the interplay of many components and configuration of those components. Complex dependencies arise and the resolution and tracking of those dependencies is essential to the reproducibility of physics results produced by these projects. MicroBooNE has temporarily solved their complexity problem by creating a simpler analysis framework, but is now expending critically scarce effort to reimplement the features of the more heavyweight framework. Eventually they will have to address the same complexity issues that they faced when they began. Some of the deficiencies appear to have originated from a lack of communication with the framework development teams, and could have been prevented through periodic visits and reporting issues to the developers.

The effort being expended to port algorithms and data products back and forth between frameworks is troubling. This porting process, especially as it is concentrated in a single individual, carries with it substantial risk both in terms of consistency and sustainability.

It was generally recognized in the review that LArSoft algorithms should be made framework-independent when feasible - although there are several caveats, it would be beneficial -- not just for porting, but also to decouple evolution of the framework from the development of the algorithms. One of the biggest advantages to having algorithms that perform specific duties independent of frameworks is the increased ability to test that they are performing properly i.e. producing a correct result. The push for framework independence from experts has this major goal testing in mind. Framework independence should also not mean avoiding use of utility and other assist libraries for developing algorithms. When algorithms are made framework-independent, they may not be able to take full advantage of important features such as inter-product references, as well as global services that manage database access, detector geometry, job configuration, error handling, and message logging. With this in mind, the granularity of the tasks performed by these algorithms should be carefully considered.

<u>Recommendations</u>

To All:

The framework-independent algorithm push needs to include testing to reap important benefits beyond use in multiple settings. Work together (MicroBooNE / LArSoft / art) on upgrades and updates that more easily accommodate multi-framework algorithms and ensure that this new direction does not cause replication of essential production framework services e.g. timing, conditions, calibration, metadata access and record. Work together to push ideas for simplifying algorithm development and data access into the core production frameworks.

Make sure that art and LArSoft team members are present at software meetings and other forums that include software activities, and are actively engaged in discussions of good

use of framework facilities and incorporation ideas that enable multiple frameworks to be used.
To the SCD:

SCD should provide help to discuss and optimize the production software configurations of the experiments' full framework software stacks. The experiments would benefit from a regular forum where specifics of the configurations can be discussed with framework experts, for example to remove unneeded services and modules from the execution.

To the Experiment:

The MicroBooNE collaboration needs to better understand its production configurations and their dependencies. If a service is not needed for a configuration, it should not be configured to run. The committee recommends that the experiment optimizes all configurations to remove unused components.

## Workflows and data management

Covered in this section are major aspects of release management, installation, configuration, and running of the simulation and reconstruction applications. MicroBooNE has a production software environment that utilizes many of the HEP standard practices for modular design, and uses many of the available FNAL-supplied software infrastructure and facilities within their software management chain.

### Packaging, Building and Release Procedures

Findings

MicroBooNE makes packaged binary releases available through the FNAL facilities CVMFS and SciSoft. They utilize releases of LArSoft, art, and important externals such as ROOT, Geant4, and GENIE as intended to ensure consistent results. It is reasonably straightforward to find and set up a "uboonecode" release on the standard interactive nodes.

Algorithms used and developed by MicroBooNE have homes in LArSoft, and development of these algorithms is accomplished using LArSoft tools and practices, and using the LArSoft repositories. MicroBooNE relies on the tagging and release cycle of LArSoft to ensure that changes show up at the higher uboonecode level, and uboonecode build and packaging steps follow periodic builds of LArSoft.

MicroBooNE reported that weekly builds are declared production releases when needed. The process itself is not transparent within the collaboration. There are no firm procedures to announce a production release within the collaboration. Collaborators are sometimes at a loss to know which release to choose for what purpose.

Comments

MicroBooNE makes good use of the binary distributions of lower-level packages provided by SCD. SCD does not support Mac OS X in a production environment; the experiment needs to take this into account. The reported X11 library dependency problems especially running off-site can be fixed following the example of NOvA, who packaged a X11 library version to be distributed with their software stack off-site.

The current practice of using the current LArSoft release schedule based on declaring a weekly build a production release may be insufficient for the commissioning phase of the experiment. Complete top-to-bottom release requests may need to be cleanly made on-demand that minimally cover uboonecode and LArSoft. Furthermore, the collaboration will need to decide and publish when these new releases are available and what they can be used for. There was not enough time in the review to know if the collaborative policies and procedures within LArSoft match the needs of MicroBooNE moving new code into production releases.

Recommendations

The committee recommends that MicroBooNE review and revise their release procedure, and ensure that production releases are cut and announced transparently to the collaboration. MicroBooNE has to document releases and their purposes.
The collaboration also has to familiarize itself with cutting releases of their software outside the weekly build schedule.

## Integration and Validation

Findings

MicroBooNE has invested effort into developing specialized integration test targets available to the build and CI systems via LArSoft. A member of the MicroBooNE collaboration was one of the primary developers of these features. These features are tied to the “v2” requirements and release of the CI system, which is still under development.

A plan for creating validated software releases was not clearly presented at the review. This includes validating physics from release to release, unit testing important algorithms and functions, and integration testing that ensure that changes in one area do not break other areas. It was explicitly stated in the summary that people need to be encouraged to write unit and integration tests, as well as to maintain them. Furthermore, no information was given about how LArSoft validation steps will be used along with MicroBooNE validation steps to ensure good releases.

Comments

SCD has reviewed and accepted the CI v2 requirements and is working on their implementation.

It is important that everyone understand when to use each of the various test targets as code is developed. It is essential that tests for any level of component be as easy as possible to add and maintain. It is desirable for tests to be added by someone other than the main author of a body of code. Those in charge of development need to work more closely with the maintainers of the build system tools and the CI system tools to be sure that forward progress is made on testing.

Platform upgrades and changes are inevitable. A recent example is SLF5 to SLF6 and the introduction of Mac OS X as a development platform. There is much concern that validation between available platforms - especially physics results validation - have not been done sufficiently.

Recommendations

To the SCD:

SCD should finish the CI v2 system implementation. The experiment should encourage unit tests and physics validation, especially across different platforms.

To the Experiment:

We recommend that the experiment implements some level of overall physics validation of the MicroBooNE software stack. We recommend that there is minimally an integration test that runs the reconstruction chain through a number of fixed events and minimally reports “worked” or “failed to work”.  We also recommend that special tests for critical algorithms and processes are designed and added to the validation.

## Configuration of Workflows and Performance

Findings

During the reconstruction phase, many competing reconstruction algorithms are run serially within the same job, which is appropriate for sharing data from early stages and ensuring the same running conditions and and can allow for direct comparison of results. Timing results from production runs show a very slow per-event rate - on the order of 500 seconds per event. Several talks requested expert help for analyzing and improving the performance of the processing steps and stages. In addition, requests were made for help with improving the custom Geant4 physics lists that currently exist.

The simulation applications require 4GB of virtual memory to run - no requirement was presented for actual physical memory. Since many of the FermiGRID nodes are configured with 2GB slots, this requirement limits where the applications can be scheduled to run. Discussions in the review indicated steps in the GENIE event generator were responsible for this requirement, but it was not clear if it was Monte Carlo data product processing or algorithms within modules that cause this higher memory use.

Comments

SCD has dedicated staff for helping with effectively configuring and using Geant4, but they cannot work independently from the experiment experts. Improvements in this area may have a significant impact on space and time requirements.

There is much expertise within SCD relevant for performance profiling, high-performance C++, and algorithms. Establishing a regular schedule for mini-reviews with SCD specialists that involve all these areas, and applying recommended changes to algorithms within the reconstruction workflow may be necessary to significantly improve performance. This interaction with SCD will also need to include examination of the memory used by algorithms and the organization of the data structures they use.

Recommendations

The MicroBooNE simulation group ought to work with the SCD simulations group to review the Geant4 physics lists and any other relevant configuration.

SCD should consider giving a tutorial on identifying badly performing code and improving the performance of algorithms.

## Production tools, file handling and Metadata

Findings

Within the physics application, there are missing SCD infrastructure services that still need to be provided, and incomplete ones that have caused MicroBooNE to provide expanded implementations. One example is the need for metadata for generated histogram / n-tuple files. To correct this, MicroBooNE has implemented the TFileMetadataMicroBooNE service.

Art-ROOT files contain a relational database which is key for the metadata handling for SAM. The APIs to manipulate entries in the relational database are insufficient for use.

MicroBooNE is using 3 metadata fields in SAM to describe collaboration-specific attributes of datasets.

A bridging product exists that defines data types. It is used both in the online and offline software environments.

There are two independent solutions proposed for running production: the larbatch solution through use of “project.py”, and the incomplete solution PUBS. The standard jobsub_client is used by the larbatch solution and has been tested on the supported environments, but there
are reported caveats involved in its use.

A well-structured configuration of a production workflow was presented at the review. The current configuration runs in segments, with each segment producing an intermediate file that is not permanently stored, but is temporarily made available through global file systems to downstream steps. Only the final data files, which accumulate all prior data products, are catalogued and stored permanently. At the same final step, final files are generated in both LArLite and AnalysisTree (a flat ntuple) format for further analysis work. The current applications cannot be configured to use local disk on batch nodes to pass intermediate result files from one step to another.

The particular use of flux files in the GENIE stages constrain the sites where applications can run. The cause of this is attributed to random access within these multi-GB files, used in an early GENIE helper event selection module. The “beam group” within the physics analysis section of the experiment provides these flux files. Little to no information was given at this review concerning the processing, storage, or use of flux data supplied by the beam group.

Comments

There is a concern that by using only 3 metadata fields in SAM, a parallel bookkeeping infrastructure has to be built to allow the collaboration to understand what is stored in specific datasets. The current bookkeeping approach uses static web pages and wikis.

Of further concern is the number of people working on the bridging product defining data types. Care needs to be taken that consistency is ensured. There might be a need for versioning of data types during data taking when backward compatibility cannot be guaranteed.

The script production workflow setup duplicates many functionalities of standard SCD systems. For example, the file handling used in the script setup re-implements the SAM file handling. This would enable the off-site usage of the MC production workflow without having to manually move files back and forth. The same is true for the PUBS setup as far as could be understood from the review. It is advisable for MicroBooNE to investigate if this duplication and the subsequent maintenance effort is worth spending.

A detailed breakdown of processing times and storage usage needs to be provided for the production processing procedure. In particular, overheads in file movement, data product reading, writing, compression, and data product re-formation i.e. serialization need to be examined carefully. Newly developed features may alleviate the need for merging and carrying data products forward from step-to-step, which may significantly reduce the data movement and processing overheads. If the new features are not adequate for the MicroBooNE workflows, then the LArSoft / art teams should be engaged to see if additional infrastructure modifications can be made to reduce overheads. There may also be ways to modify the workflow to reduce the movement of data files through network storage and operate from step-to-step using local storage.

SCD is currently working on an improved scheme for distributing large auxiliary files like flux files. Although MicroBooNE claimed that FermiGrid resources are sufficient for their MC production needs and they don’t need a generic solution for flux file distribution There are also efforts to remove the need for random access to 100+ GB of files in GENIE.

There are concerns though that the production of the flux files was not mentioned when MC production needs of the collaboration were presented. We encourage MicroBooNE to include beam MC into their planning procedures.

Recommendations

to SCD:

SCD needs to investigate missing metadata handling in art/LArSoft and integrate MicroBooNE specific solutions into the general code base.

to MicroBooNE:

Review how the collaboration can discover information about the content of datasets, trying to avoid duplication of metadata systems.

## Event Display

Findings

MicroBooNE presented the currently used event display solution, from a full framework integrated version to a lightweight web-based solution.

Comments

No event display solutions is ever feature complete. They tend to be further developed during the full lifetime of an experiment. MicroBooNE has request to improve the event display solutions, they are not critical though.
The web-based event display solution is targeted to be used in the control room during data taking. Of concern is that it is only supported by a single PostDoc. Security audits were done, but not by the security group.

Recommendations

MicroBooNE is recommended to make a plan for long-term support of the web-based event display solution. If the web-based event display is going to be used outside the control room network, meaning accessible freely from the internet, a security audit by FNAL security experts needs to be conducted. Also a critical review of the web-based event display's impact on the computing infrastructure, especially tape staging, needs to be conducted.

## Data taking

### Commissioning/Online/DAQ

Findings

MicroBooNE presented an intricate and complicated plan for their online computing infrastructure. Various pieces are in different states of implementation, integration and testing. A coherent plan was not presented: prioritization of individual tasks and manpower estimates were not described. The importance of certain items were mentioned but no structured plan to realize these plans were presented.

Comments

The lack of structured planning and effort assignment is of concern and might put the start of data taking in jeopardy. Although stated by MicroBooNE that delays are not impacted by accelerator schedules, they would have impact on the overall goal of MicroBooNE to collect 6.6E20 POT in the currently allotted running time and the overall physics harvest.
Of major concern is the state of the Huffman coding in the FPGAs. MicroBooNE presented it as critical to reduce the bandwidth out of the warm electronics by an order of magnitude, allowing MicroBooNE to run at the design data taking rate. This step needs ironclad validation to make sure that data are not corrupted at an early step in the DAQ. No evidence was given that an order of magnitude reduction can be achieved.

The overall online workflow treating data coming from the detector is very complicated and sometimes confusing. Multiple copies of the RAW data are passed back and forth through the system. Of note, it is not clear if the binary data copy, the first RAW data copy, is being compressed in a dedicated step in the online workflow or if compression in enstore during the tape write has to be activated. This needs to be clarified with the FNAL facility team.

Also the design and implementation of calibration workflows was not described in enough detail to see a coherent plan. Effort is needed to set up calibration workflows, and people need to be identified to be responsible for procedures and schedules for calibrating physics-quality data.

Although discussed, it was not clear in the end if the online system is sufficiently protected of network cuts and can operate autonomously. Especially the database setup and dependencies on other databases like IFBeam needs to be checked and made robust. We also noticed that no streaming PostgreSQL replicated is setup for the DAQ cluster. A concrete question arose concerning the offline database (ifdb01): can MicroBooNE take data without access to it?

It will be very important to load-test the full system that will be operating online, as well as subsequent offline processing steps.

No plan was presented how to practically organize data taking, if there is a need for pager carriers and who are the responsible people for online software. A general question is: how many people are needed on shift at the beginning and how are shifts organized?

Recommendation

The experiment should put together a coherent schedule and big picture plan for the pre-commissioning work, ensuring effort is available and not double-booked, with appropriate contingencies.

We recommend to review the overall online workflow and simplify where possible. Care should be taken that the workflow can be operated when the network connection to the FNAL computing facility gets cut. This is especially important for access to databases like IFBeam and others.

We strongly recommend to hand over the admin. responsibilities of the online DAQ databases to FNAL CD as well as setting up appropriate PostgreSQL replication in the DAQ cluster for redundancy.

We recommend MicroBooNE to develop a plan for shifts and alarming for data taking.

## Swizzling (Raw to Root Conversion)

Findings

The collaboration plans to perform the first stage of processing of their raw binary data into an art-root format through a process referred to as “swizzling”. The experiment plans to run these jobs within their DAQ cluster. Specifically they plan to use computing cycles on their secondary event builders (SEBs) and on their primary event building nodes to institute a condor batch queue that will accept “swizzing” jobs. These DAQ resources are critical to the operation of the data acquisition. The experiment plans to perform performance benchmarking of this process.

Comments

There is significant risk inherent in the experiment's plan to use critical DAQ resources for a non-critical and non-time critical piece of nearline/offline data processing. In particular the current design presented by MicroBooNE will need to use significant IO resources to transfer the data files between the different storage and compute nodes, will use significant local I/O and computing resources to process the data. The design does not make specific provisions to protect their DAQ's data flow and event building from the resource that will be consumed by these nearline jobs nor from non-standard flow conditions that these nearline jobs may provoke.

The full range of problems that may be encountered in trying to implement and deploy this computing model can not be easily captured with simple performance testing, and will not be captured or evoked even during initial DAQ testing. Past and currently running experiments have extensive experience in designing and debugging systems for nearline data processing and their experience has repeatedly avoided the use of DAQ computing resources and nodes for non-critical applications based on the encountered risks to DAQ stability.

Due to this extensive experience we believe that the MicroBooNE design for raw to root conversion presents a high risk to the stability of the data acquisition system and may adversely impact the ability of the experiment to meet their commissioning goals and timelines.

Recommendations

The MicroBooNE experiment should abandon their plans to run their "swizzling" jobs on any platform or resource that is critical to or may affect their data acquisition chain. The experiment should instead identify or specifically allocate, dedicated computing resources to these jobs. These resource can be deployed to the experiment's computing center but the resource and workflows should be arranged to minimize the potential for the system to affect DAQ stability.

If the MicroBooNE experiment can remove any time criticality of the "swizzling" process, then the experiment should move this processing to a purely offline environment and perform the required "keepup" processing in a mode that is sustainable or can be offloaded SCD supported operations groups.

## PUBS / production system

The MicroBooNE experiment is in the process of developing a workflow management system to handle in an automated fashion the advancement of data through the MicroBooNE data processing/calibration/analysis chain.

Findings

The MicroBooNE experiment has adapted work that was done for the Double Chooz experiment's processing framework to develop a database based workflow management system that they have named PUBS. The experiment is currently in the process of deploying this system to their DAQ/Nearline computing systems. The system is intended to handle the movement of data between DAQ systems, the initiation of jobs to perform the merging of DAQ data with beam spill information, the initiation of jobs to perform data format conversion, and other tasks and their dependencies which copy or move the data to storage resources. When this system is complete the MicroBooNE experiment plans to expand its usage to their offline computing and production needs.

The PUBS system is based on a database state machine. The development of this database is separate from the other work that is be done for the databases used in the DAQ, calibration and offline. It is intended to be separated and isolated from the data taking and to not provide a bottleneck to the data taking or commissioning. However, the PUBS based workflows are the source of the DAQ monitoring and Data Quality Monitoring (DQM) which ties this service intimately to the ability of the experiment to performing commissioning tasks.

The MicroBooNE experiment presented a workflow management system that was not complete and which required non-negligible effort to develop and integrate core components of the system and develop essential workflows that are required for commissioning. This work was estimated by the experiment to constitute greater than 12 weeks of effort. A limited timeline and identification of effort was presented. The support and maintenance of the core portion of this system is currently handled by one Postdoc.

There was a desire expressed by the experiment to have the SCD take over operations of this system in the future through the Offline Production Operations group.

Comments

The MicroBooNE experiment has recognized that there is a definite need for a workflow management database which is able to organize and track the flow of data through their processing chain. However, there is significant functional overlap between what the MicroBooNE experiment is developing, and the well established tools and services that the SCD provides and supports. In particular many of the core functions of the proposed PUBS system which duplicate the SAM data management system, the Fermi File Transfer Service (F-FTS) and the standard jobsub infrastructure for job submission.

The data flow and processing paths that the experiment presented as being part of the initial PUBS domain (and residing within or near the DAQ environment) were much more complex than what other (similar) experiments have used to accomplish their DAQ/online monitoring, data quality monitoring, nearline processing and archival data storage. The schematic diagrams that were presented in both the Online/DAQ presentation and the Data Management presentation were both confusing and inconsistent with each other. There are also a number of different services that is not clear are fully developed, integrated or tested. Given the limited manpower resources that are available to the MicroBooNE experiment, there appears to be significant risk associated with timely completion and integration of the (as presented) PUBS system and associated infrastructure.

The other concern that the committee has is that the PUBS infrastructure is being developed by a limited number of individuals whose technical skills, knowledge and expertise may not reside within the highly specialized domain of large scale data management. In particular the experiment may not have the sufficient experience in working with the archival storage facilities that are provided by the computing division and may not be aware of the extensive experience and expertise that resides within the SCD related to data management for Run II, CMS and NOvA and other high data volume experiments.

The development of separate toolsets by the experiment risks isolating them from the expertise and support of the common, robust and proven tools that the SCD has developed.

Recommendations

The MicroBooNE experiment should review their data and workflow model as it pertains to the DAQ, Online monitoring, nearline and archival storage. In particular the experiment should look to simplify the number and complexity of paths that they currently are using and the [excessive] data movement that is occurring within the DAQ environment. They should examine similar data flow and processing models that have been used successfully for other experiments, and should adopt similar patterns and models. In particular the experiment should look examine models that reduce the latency in producing DQM metrics and verifications that the DAQ as a whole are functioning properly.

The experiment should engage with expert members of the SCD to map their data processing needs, wherever possible, to the established and supported data management, data movement and job submission tools that are provided by the SCD. The experiment should engage SCD resources in deploying these solutions. In particular, the MicroBooNE experiment should use the F-FTS system to register and move ALL critical data to durable and archival storage. It should similarly rely on the F-FTS system for the proper registration and cataloging of the data with the SAM data catalog. The experiment should then use the SAM analysis project infrastructure to organize, perform and monitor their data processing tasks.

The SCD should engage with the MicroBooNE experiment to collect their needs and requirements for an offline production database system that is well integrated with data handling and job submission tools. The SCD currently has a project that is developing a system to meet the the generalized needs of the running experiments at Fermilab and can bring significant development effort and expertise to the project. MicroBooNE should become a principal stakeholder in this system and should be engaged in its designs.

## Analysis readiness

Findings

MicroBooNE demonstrated MC-based analysis capabilities and has all pieces of the end-to-end analysis workflow in place. Collaborators successfully analyzed MC samples following available documentation and with the help of their co-collaborators.

Comments

Best practices for reproducibility and coherent usage of tools/algorithms/configurations are not followed in general in MicroBooNE.

Recommendations

We recommend to review analysis practices in MicroBooNE with emphasis on coherence and reproducibility of the analysis workflow and write down best practices before or soon after start of data taking to guide the collaboration and ensure reproducible highest-quality physics results.

# 2. Engaging non-expert resources and best practices

### Are the tools, infrastructure, and established processes sufficient to engage non-expert resources from the collaboration

Findings

MicroBooNE enables non-expert collaborators to perform analyses using a combination of SCD and collaboration written frameworks, tools, and infrastructure. MicroBooNE offers a choice of three frameworks: art/LArSoft, LArLite and AnalysisTree (flat-ish ntuple). MicroBooNE leverages and is closely integrated with LArSoft. Sufficient documentation exists to ramp up analyses and development efforts. Collaborators are supporting each other in their efforts.

Comments

The suite of choices is sufficient for collaborators of different programming skill levels and applicability to analysis tasks to be productive. Despite LArSoft being effort limited, their use of LArSoft appears to be successful by a large fraction of the collaboration developers. Support and education of MicroBooNE collaborators in using computing-specific tools like SAM are lacking compared to software-specific tool support.

Recommendations

The experiment should engage in efforts to educate their uses in the proper use of the SAM data handling system for the retrieval and analysis of data, as well as in other computing systems used by the collaboration.

## Are best practices employed in these processes?

Findings

The tools used by the majority of the analyzers are lacking fundamental features like provenance tracking, strict versioning, and release procedures. Validation of pieces of the whole workflow chain up to analysis exist but validation is not done in a systematic way in regard to guarantee reproducibility of physics results.

There is very limited guidance for the collaboration regarding which software release to use for development and analysis. The experiment produces weekly releases of LArSoft and MicroBooNE libraries but the process to declare one of the weekly releases a production release is not transparent for the collaboration.

Comments

There is a significant functional overlap between tools developed by SCD and tools developed by MicroBoone (e.g. PUBS & SAM, LArSoft & LArLite). Furthermore, there are at least three copies of the data (art-root format, AnalysisTree format, LArLite format) being produced and translation of data and algorithms between each. Care must be taken to avoid mistakes.

The best practice of separating algorithms from framework specific code improves portability and testability. These efforts are encouraged.

The rise of lightweight organically written alternate frameworks, like LArLite and AnalysisTree, occurs in nearly every experiment with support from a typically dedicated and passionate set of experiment collaborators. MicroBoone should realize that adoption of such frameworks carries risks. As those who maintain and supply help for those frameworks move on, support holes may be created. There is no guarantee that the SCD will be able to supply effort to backfill support, especially in this era of constrained budgets and effort.

Recommendations

MicroBoone is encouraged to develop a support plan for analysis in the three frameworks including how to guarantee reproducibility of analysis results and how to increase transparency for non-expert collaborators, perhaps in collaboration with other experiments.

## 3. The personpower needs and availability, both from the experiment and SCD.

Findings

The presentations from the collaboration suggest that MicroBooNE has sufficient personpower in the collaboration to analyze the data and extract physics results. It is unclear what the exact distribution of effort is within the collaboration. Critical tasks such as online software development, testing and validation, and database development are staffed at a much lower level than analysis. The online software box in the organization chart is empty. A small number of collaborators are multiply represented in the organization chart and have multiple critical responsibilities. There have been episodes of communication issues between collaborators.

SCD is involved in many aspects of software and services for MicroBooNE and overall the personpower situation seems appropriate. On the other hand, the LArSoft project is limited by personpower and is unable to address all high-priority items on its list.

Comments

Lower levels of staffing for critical tasks create concern over schedule delays, particularly if a task takes longer than expected and the responsible person is also responsible for doing other tasks. Of concern was also that no data quality monitoring tools were demonstrated beyond the event displays. This needs to be demonstrated before the start of data taking and may require the direction of additional personnel to these tasks. The staffing of the LArSoft project has impacts on all of the LArSoft stakeholders: MicroBooNE, ELBNF, LArIAT, and LAr1ND.

Recommendations

Spokespeople should audit and enumerate the effort from the collaboration on critical tasks. They should develop a schedule for commissioning that spreads effort among collaborators that addresses critical-path tasks, and which identifies the resources needed to accomplish the tasks. Interfaces and communication within the collaboration and between collaborators and SCD must be monitored for breakdown. Regular communications between collaborators at Fermilab and at remote institutions are needed to ensure that collaborators are engaged and are working on appropriate projects that meet the needs of the collaboration.

SCD should review the adequacy of the staffing of the LArSoft team, and seek ways to direct additional effort to it. Close collaboration between the LArSoft project and its stakeholders is strongly encouraged.

**Questions to MicroBooNE submitted 2/23/2015 6 PM CST**

- Please give us an overview timeline of commissioning and data taking periods over the next 3 years.

- It was not clear to us which tasks which are required to be ready for data taking are planned and which are complete.
    - Please list planned tasks with a timeline and effort estimate, including especially the integration, validation and load-testing steps.

- By when does MicroBooNE need to access flux files off-site?

- What is the experiment's plan for evolving LArLite and AnalysisTree? What is the experiment's policy on how LArLite and AnalysisTree are supposed to be used by the collaboration?
    - What is the experiment's effort distribution between work on LArSoft, LArLite and AnalysisTree?

- We didn't see a concise statement about the benefits and shortcomings of LArSoft and art from the MicroBooNE experiment perspective. What are usability improvements to art and LArSoft themselves that could make a big impact to MicroBooNE?