Intensity Frontier: Software and Computing Outlook

Clark McGrew
Maxim Potekhin
Brett Viren

HEP Software Foundation Workshop
January 20-22 2015
SLAC

Intensity Frontier a FNAL and beyond

- FNAL is a pre-eminent center for accelerator-based neutrino physics in the US. A few current experiments:
  - NOvA
  - MicroBooNE
  - MINERvA

- ...and also:
  - muon physics:
    - mu2e
    - g-2

- A sample of a few other recent, future and R&D projects relevant to IF in the context of HSF:
  - Daya Bay
  - T2K
  - CAPTAIN
  - LBNE

Fermilab

- Every National Lab is organized and operates differently, with varying Software and Computing (S&C) requirements, capabilities, scale and type of facilities and effort profiles. Some labs do have dedicated S&C divisions and some don't. FNAL stands out because of sheer scale of S&C work that's being done there, and in particular the role of its Scientific Computing Division
    - ❖ development and maintenance of physics software tools from data acquisition to simulation to analysis frameworks
    - ❖ facilities, operations and workflow management
    - ❖ storage and metadata
- Fermilab is fairly unique in that it provides a fairly complete suite of services and support (along the lines presented above) to a substantial number of experiments. *Sometimes a vast portion (if not all) of software design, development, deployment and operation takes place within the FNAL perimeter.*
- Often, local computational resources (e.g. Fermigrid) are enough to cover the needs of an entire experiment – cf. MINERvA as reported at the FIFE meeting 2013, and lion's share of LBNE computing in 2012-14. Extrapolation is tempting but may be misleading.
- *Sometimes, the downside of taking advantage of software developed and maintained at FNAL is reliance on specific tools and procedures which are site-specific and difficult to port to other sites and platforms (as experienced in LBNE and other experiments such as CAPTAIN).*

Daya Bay

- Software Strategy: steal before write. Evaluated existing software, adopted what seemed to fit best and then spent our effort adapting and extending it as needed.
- Framework: Gaudi (LHC). Made an effort to cultivate a cross-experiment Gaudi user community. Good participation by Gaudi developers.
- Build System: CMT, LCGCMT and custom orchestration script. Usual build tools (make, cmake, gcc), Python, MySQL, GSL, sqlite, Boost, CLHEP, ROOT, Geant4.
- Database Interface: MINOS DBI.
- Data Movement: SPADE system from IceCube.
- In-house development of a few critical components – either reflecting unique characteristics of the detector, or aiming to optimize performance.
- Avoided deployment on the Grid – evaluated the scale of computing resources needed, and made a calculated bet which paid off.

CAPTAIN&T2K

- **CAPTAIN: Cryogenic Apparatus for Precision Tests of Argon Interactions with Neutrinos** - important step towards future LAr-based experiments.

  Software leveraged from T2K

- T2K: "Tokai **to** Kamioka", neutrino oscillation experiment.

  Build and Configuration System, CMT-based automated build and release management, with long-term reliability and stability in mind.

  Release validation.

  ROOT TGeoManager.

  In-house job definition.

  A collection of easy to manage tools as opposed to an integrated environment.

LBNE

- LBNE plans **emphasize a truly distributed computing infrastructure** with highly portable and flexible software components shared across a large international organization – obviously, very much work in progress.
- Utilizing "art" analysis framework (developed at FNAL), and art-daq for data acquisition in 2015 (the 35t Liquid Argon prototype).
- LArSoft application suite for Liquid Argon detectors – development ecosystem and user community at FNAL.
- GEANT4-based beam simulation effort.
- Priority work items:
    - "Portable build": implementing build and configuration tools to make principal software components (e.g. LArSoft/art) easier to build from source at participating research centers and create a productive development environment – close to completion
    - Release management and Continuous Integration (cf. Jenkins)
    - Geometry description – crucial tool in the present R&D stage; developed a prototype toolkit. Emphasis on a "neutral" representation which can be exported to multiple target applications.
- Grid deployment – ran on ~20 OSG sites with CVMFS as software provisioning mechanism. **Great support from the Open Science Grid**.

Summary

Items of interest to IF on the HSF site (guidelines and questionnaire, typeset in black):

- The IF experiments would benefit from high quality software build, packaging and release management tools (re: building and testing infrastructure) particularly in cases where there is synergy between IF and non-IF projects.
- Peer reviews: undoubtedly a benefit. Same goes for consultancy.
- Access to computing resources on many platforms and architectures. In our interpretation, this can be broken into three parts: (a) "unconventional resources" – many of current and future experiments won't be able to invest sufficient effort in R&D aimed at utilizing resources such as accelerators, GPUs etc – this is where HSF can make a difference by pooling expertise and R&D from many contributors. (b) providing CI on platforms not readily available to a particular IF project. (c) sharing HPC expertise.

Thoughts on items not explicitly presented on the HSF list:

- Despite some reuse, the are no common geometry description/modeling and visualization tools. HSF could serve as a forum to discuss and establish common interfaces and share code.
- Data handling tools may not be a category abstract enough for the list, but they are so ubiquitous and central to design and performance of systems that maybe they deserve an honorable mention. Same goes for database technology (e.g. noSQL).
- Looking at the experience of the Open Science Grid – focus on continuous integration and packaging of the code really paid off. The HSF (as an umbrella organization) could help select and actively manage a few software products most promising from reusability standpoint, in a similar manner.

A backup slide

- Looking at the experience of the Open Science Grid – focus on continuous integration and packaging of the code really paid off. The HSF (as an umbrella organization) could help select and actively manage a few software products most promising from reusability standpoint, in a similar manner.
- The IF experiments aren't often sufficiently invested in, or proficient with usage of modern distributed Workload Management Systems.