Production Simulations & Analysis Update

Dylan J Temples

3 August 2022

MAGIS Science & Simulation Meeting

Outline

- Version control
- Production simulations: status & path forward
- File format & standardized header for simulations
- FNAL infrastructure & considerations
- Action items & open questions

Action items & open questions from 2022 Simulations & Analysis Workshop:
https://docs.google.com/document/d/1nqsH9k37aY4-C745IOl6FYmzIwukMOyB3zTuzAkVsSA/edit?usp=sharing

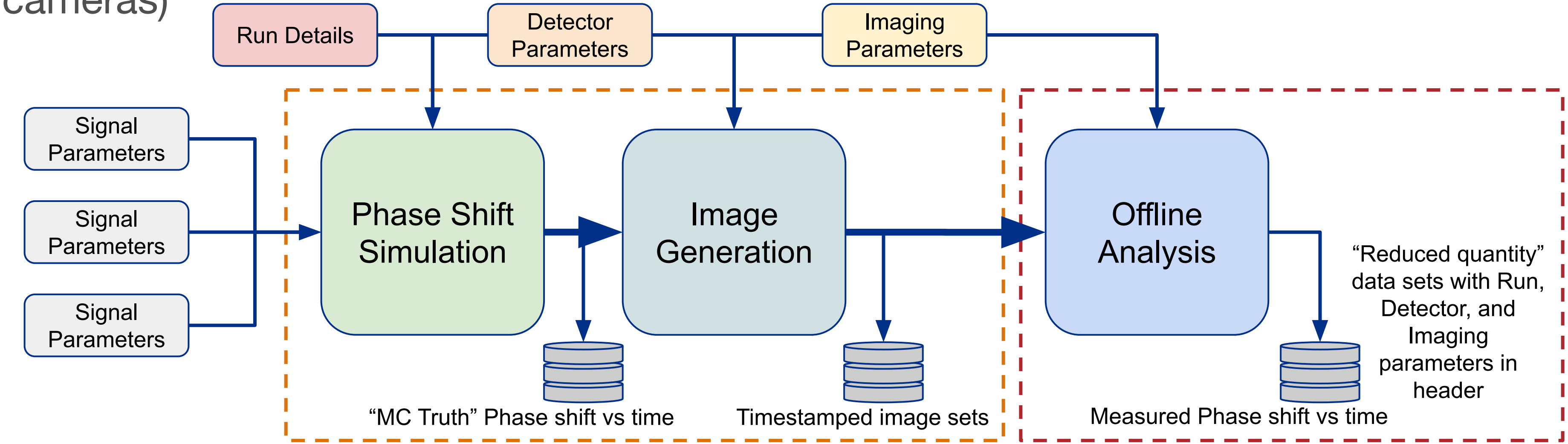
Summary report:
https://nd-docdb.fnal.gov/cgi-bin/sso/ShowDocument?docid=1527

Version Control Options

FNAL Redmine	AD GitHub Enterprise	GitHub Team / GitLab Group
In use (Sims and RPi control) FNAL accounts required	FNAL accounts required Seats limited, FNAL-managed ($) No experiment-wide account	No FNAL account required Free & paid options available GitLab: more features on free plan
No CI/CD No repository groups/structure	CI/CD enabled	CI/CD available, more capabilities in paid subscription
Paired with wiki for documentation	Online documentation pages	Online documentation pages
Available on FNAL VMs	Available on FNAL VMs	(Should be) Available on FNAL VMs
Available on AD Controls Network	Available on AD Controls Network	Not available on Controls Network
https://cdcvs.fnal.gov/redmine/projects/magis100/repository	Waiting list to get accounts on this system - probably hesitant to pay for all members of the experiment	GitLab: $19/user/month ($228/yr) GitHub: ~$4/user/month ($44/yr) https://github.com/pricing https://about.gitlab.com/pricing/self-managed/feature-comparison/

Production Simulations and Analysis

Production simulation: pick some set of signals with chosen parameters, simulate the phase shift, detector effects, up to creating images (in same format as real cameras)



Production analysis: starting from the images, fit the patterns and extract a phase shift. Should be able to run on both simulation and real data with minor user input. These algorithms used in both Online and Offline processing

Production Simulations and Analysis: Python Tools

Case study notebook: (image generation) up to date with latest version on DocDB

- Simulations, fitting using example parameters for atom cloud in imaging region

Scalar DM phase shift simulation: in final stages of testing, bug fixes & optimize

- Given DM parameters, simulate the induced phase shift over the course of a data taking campaign
- Packaged in Docker image and deployed on FNAL Wilson Cluster for benchmarking
  - Single core: 1 Campaign (1e8 images): 30 seconds, 3 GB output (no images)
    - 10 Hz cadence, ~4 month run
  - No optimization/parallelization, working on better data format/reduction to minimize disk space usage
  - Mathematica: can't do more than 1e7 images on local machine, 1 minute for 1e6 images (~100 minutes for 1e8 images)

Vector DM phase shift simulation: in progress, starting with B-L coupling

Production Simulations and Analysis: Path Forward

My work so far: Python - What alternatives should we consider?

Python. Package management is now more of a pain at FNAL. Less institutional knowledge on deploying code on VMs/Grid compared to ROOT

- Jupyter notebooks running on remote servers (e.g. VM/grid) can access on-site data and use standardized python backend analysis methods
- Methods in other languages may be importable to remote notebooks: https://www.geeksforgeeks.org/how-to-work-with-julia-on-jupyter-notebook/

Julia. NU+SLAC putting a lot of work in on this front. Focus collaboration effort here?

ROOT. (C++) Lots of support and knowledge at FNAL, format can be analyzed also in Python and Mathematica (possibly Julia). Steeper learning curve than Python, but easy to deploy code at FNAL.

Mock data challenges: once production simulation hits the first "stable" version, produce a mock science-run dataset - make available to collaborators to go signal hunting.

- Exercise production analysis tools, in advance of real data
- Analyzers get used to data format, can develop their own tools (later folded into production analysis)
- Identify gaps/bugs in simulation tools

Other points: Docker images for deployment? Single target OS for production code? (Fermilab Scientific Linux: CentOS 8)

File Formats and Simulation Output Headers

How do we want to store simulation and reduced quantity data? Excluding images

Options: ROOT, hdf5, raw binary, Julia-specific file formats?

Select a portable file format - easy to load in multiple languages

Simulation & data headers

- Identified fields to record: values of constants used, run details, detector & imaging parameters, signal parameters [Steve/Jeremiah/Dylan]
- Header read/write tools implemented in Mathematica (binary) and Python (hdf5)
- Need to interface with NU/SLAC regarding their Julia simulation development

FNAL Computing Infrastructure

Currently, one VM set up: magisgpvm01.fnal.gov - all MAGIS-100 collaborators with FNAL accounts should have access (need the VPN).

Wilson Cluster: https://computing.fnal.gov/wilsoncluster/

- Test cluster for things that will eventually be deployed to FermiGrid
- CPU-only and GPU-accelerated nodes (limited)
- MAGIS-100 has access and storage space allocated - if you have VM access, should be able to log into WC
- Deployed Python DM simulation code in Docker image for rough benchmarking

FermiGrid:

- MAGIS-100 users have access already
- Nothing tested on grid yet - no job submission scripts yet

Computing division liaison: someone to help set up software, create setup & job submission scripts, provide support, etc

Relevant Action Items & Open Questions

1. **What software packages do we want available on the FNAL virtual machines?** Python, Julia, others?
    a. **Can we use the free Mathematica licenses on each RPi to run simulations or analysis?** We have “spares” - could set one/few up as a Mathematica machine for those without institutional licenses

2. **What scales of simulation output will be stored on LTS: images vs RQs?** Images take up significantly more space, will need cropping/compression. If we have standardized fitting routines, we don’t need to save the images from simulations

3. [Complete] **Develop a standardized header for each simulated dataset to document its contents.**
    a. Steve, Jeremiah, Dylan were working on this - need to interface with NU/SLAC simulations

4. [Complete] **Investigate experiment-account for AD Enterprise GitHub.**
    a. No “experiment”-wide account. Users can be added to a “Team” but FNAL pays for this through AD, may not be keen on giving all collaborators accounts. Requires FNAL accounts for all users.
    b. **Agree on version control solution and act.**

5. [In-progress] **Deploy a first-pass simulation to the FNAL grid to benchmark performance and inform computing/data-storage needs going forward.**
    a. Docker images deployed to Wilson Cluster, DM simulation run: 1 campaign ~ 30 seconds (single core, no optimization)
    b. Large output file sizes (phase & time, w/ diagnostics: 30 MB/1e6 images) - need to work on data format to decrease this (one campaign ~1e8 images)

Relevant Action Items & Open Questions

6. [In-progress] **Identify which existing tools should be used for production code, and what is yet to be developed.**

7. [In-progress] **Owners of existing code should interface with owners of code with similar purpose to potentially collapse them into a single tool that can be widely used for the same purpose (everyone uses identical algorithms):e.g., phase extraction, fitting, atom diffusion.**

8. [Not started] **Reach a consensus on development standards for production code (issue-branch-develop-merge pipeline, requirements of unit tests, etc.).** There may be a document on the DocDB aboutcode standards for Case Study, may want to revisit this in context of production simulations, but may be too specific to Mathematica.

9. [Not started] **Begin planning the schedule for mock data challenges.**

10. [In-progress] **Developers/maintainers of institutional code should ensure a version of their code is hosted online and its location, basic usage instructions, and contact info is relayed to Jeremiah to be put on Wiki.**
    a. Identify areas where your code can use external help.
    b. Identify where your tool would fit in an end-to-end simulation. What are the interfaces?

11. [In-progress] **Select a language for production simulations and analysis.**

12. [Not started] **Get a computing division liaison for the experiment.**

Thank You!

Backup & Notes

MAGIS-100

U.S. DEPARTMENT OF ENERGY | Office of Science

Gordon and Betty MOORE FOUNDATION

UKRI Science and Technology Facilities Council

THE KAVLI FOUNDATION