Data & Analysis Preservation: status update

Maxim Potekhin
Nuclear and Particle Physics Software Group

Brookhaven National Laboratory

PHENIX DAP Meeting
10/14/2021

Overview

- Maxim has joined sPHENIX so it’s 0.5 FTE between PHENIX and sPHENIX
- HEPData
- Website
- EMCAL Data and Analysis Preservation
  - Archival of the data component in the mass storage (gpfs)
  - Preservation of code in the PHENIX repository on GitHub
  - Workflow capture on the web site (detailed notes)
  - Initial template for some of the necessary REANA workflows

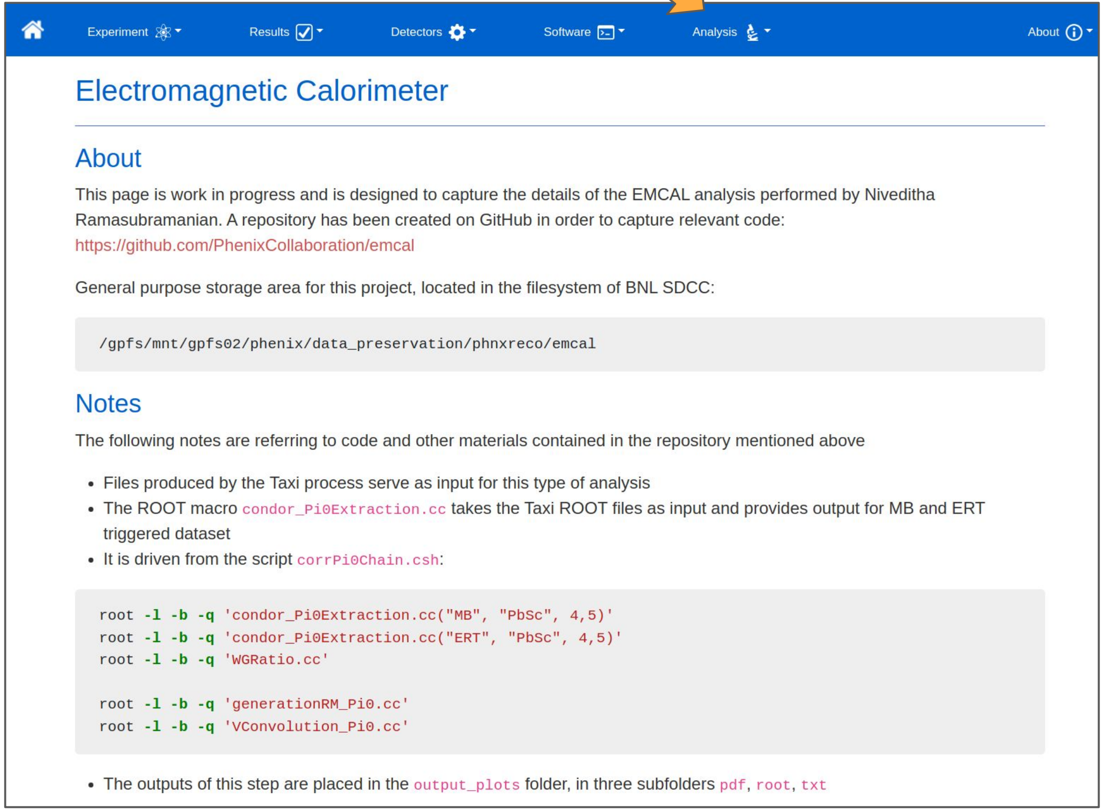
HEPData

- Ongoing activity, the master spreadsheet has been updated
- ppg147 and ppg003 have been published
- ppg241 is in the pipeline and close to completion
- People didn’t have enough time to address a couple of other items

Website

- More conferences/Zenodo links added (keywords + conference page)
- Added a new page on EMCAL analysis, linked from the “analysis” menu
  - Documented location of the “data preservation folder” and the new GitHub repo
  - Started annotating Niv’s logic from the slides presented in the last meeting, work in progress

Website - the new page



Electromagnetic Calorimeter

About

This page is work in progress and is designed to capture the details of the EMCAL analysis performed by Niveditha Ramasubramanian. A repository has been created on GitHub in order to capture relevant code: https://github.com/PhenixCollaboration/emcal

General purpose storage area for this project, located in the filesystem of BNL SDCC:

```
/gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal
```

Notes

The following notes are referring to code and other materials contained in the repository mentioned above

- Files produced by the Taxi process serve as input for this type of analysis
- The ROOT macro `condor_Pi0Extraction.cc` takes the Taxi ROOT files as input and provides output for MB and ERT triggered dataset
- It is driven from the script `corrPi0Chain.csh`:

```
root -l -b -q 'condor_Pi0Extraction.cc("MB", "PbSc", 4,5)'
root -l -b -q 'condor_Pi0Extraction.cc("ERT", "PbSc", 4,5)'
root -l -b -q 'WGRatio.cc'

root -l -b -q 'generationRM_Pi0.cc'
root -l -b -q 'VConvolution_Pi0.cc'
```

- The outputs of this step are placed in the `output_plots` folder, in three subfolders `pdf`, `root`, `txt`

EMCAL

- Created a new folder in the “data_preservation” area established a while ago, under the user “phnxreco”, to preserve Niv’s initial data: /gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal
- Created a new PHENIX/GitHub repo for the code: https://github.com/PhenixCollaboration/emcal

REANA

- Initial simple templates created for two of the steps of the EMCAL analysis
  - *“Block 1, MB and ERT datasets”*
  - *“Block 2, creating histograms”*
- Testing the basic layout of the directories and upload to the server
- Added to the “reana” repository of the PHENIX organization on GitHub

REANA

```
version: 0.0.1
inputs:
  files:
    - /gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_0.root
    - ./secNtuples.csh
    - ./secondaryNTuples.csh
    - ./Pi0EmbedFiles.C
    - ./Pi0EmbedFiles.h
    - ./DeadWarnRun16.txt
    - ./timingDeadWarnRun16.txt
workflow:
  type: serial
  specification:
    steps:
      - environment: 'registry.sdcc.bnl.gov/sdcc-fabric/rhic_sl7_ext:1.3'
        commands:
        - mv gpfs/mnt/gpfs02/phenix/data_preservation/phnxreco/emcal/Pi0/middle/simPi0_0.root pi0_dAuMB.root
        - chmod +x ./secNtuples.csh
        - ./secNtuples.csh > output.txt
outputs:
  files:
    - output.txt
```

REANA - data upload

- A file, a number of files with fully defined names or a whole folder can be specified in the job submission YAML file as inputs

- However, the file names cannot be given as parameters at submission time i.e. they are practically hardcoded in YAML
  - This is currently not possible: reana-submit -env file1=pi01.root file2=pi02.root, with variables “file1” and “file2” referenced in the submission YAML file
  - This presents a problem for analyses involving large numbers of various files

- Solutions:
  - Auto-generate YAML files (already practiced doing that) - need to manage data products...
  - Use XRootD to dynamically upload files from jobs running within REANA
  - Use REANA capabilities to generate complex workflows i.e. upload folders at once an let REANA jobs process the data as needed, wrapping up processing in less steps

REANA - issues

- The overall EMCAL workflow is complex so structuring it takes effort
- The original code relies on *many parallel Condor jobs* in one of the steps
  - Some of the logic needs to be rewritten as we can’t use Condor in REANA
  - Optimal solution - use DAGs (including **parallel execution**) in REANA to create a corresponding workflow description
- It’s a different syntax/setup from what we’ve been using so far (just linear)
- + it takes care of bookkeeping necessary for this to work and optimizes use of REANA
- – parallel workflows require complex YAML syntax which is a bit of a learning curve
  - Two options - Yadage and CWL schemas for describing workflows
  - Will take some learning curve to master

REANA - plans and priorities

- There is more value in complete documentation and clear code than just getting it to run as a black box
- More material needs to be developed for the website
- Cleanup of the code (and perhaps using more descriptive names for files and macros) would be a good idea
- Need to understand how to run complex workflows in REANA
- Certain components may be amenable to publishing on Open Data even on a medium time scale, complete with data and code (if we agree on that)