Proposals on Analysis Preservation

(According to Sebastian's talk on LHCb Analysis&Software Week)

https://goo.gl/ngAzhn

Mingrui Zhao[1]

Updated: 08/14/17

CEPC Simulation and Software meeting

Outline

1. Introduction
2. Analysis preservation and in CERN&LHCb
3. Draft suggestions for CEPC analysis
4. conclusion

Motivation for analysis preservation

- Reproducibility is a fundamental scientific requirement.
- HEP has special responsibilities, due to large/long term projects.
- HEP AP addresses several problems of knowledge transfer:
  - Collaborative working
  - Knowledge preservation and during review
  - Knowledge transfer to other analysis teams
  - Knowledge transfer to future generations

Publication policies

Nature: authors are required to make materials, data, code, and associated protocols promptly available to readers without undue qualifications.

Status

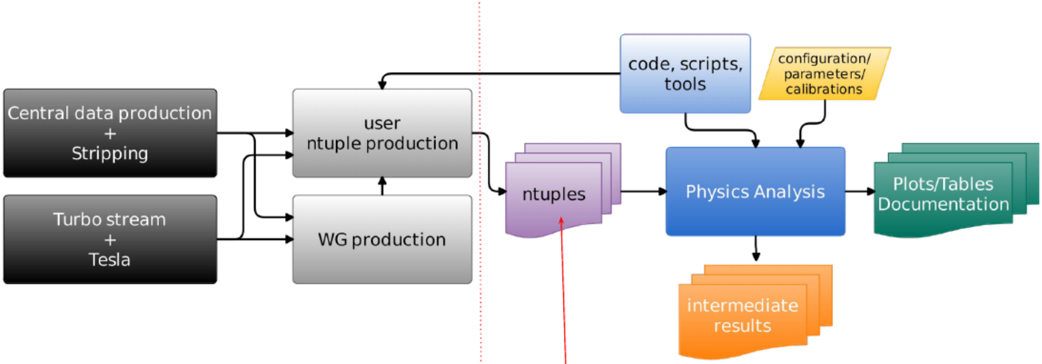
- Analysis preservation is NOT something naive and trivial.
- Usually painful to repeat the analysis.
- Where does the ntuple come from?
- Which version of the software do I use to produce the result?

Domains for Analysis preservation

Preservation = Automatically rerun analysis

- Analysis repository: analysis tools(code), logic, version
- Analysis pipeline: analysis steps
- Runtime environment
- Input data storage

Scope of analysis preservation in LHCb



Scope limited by resources to reproduce input data (MC/reco/stripping)

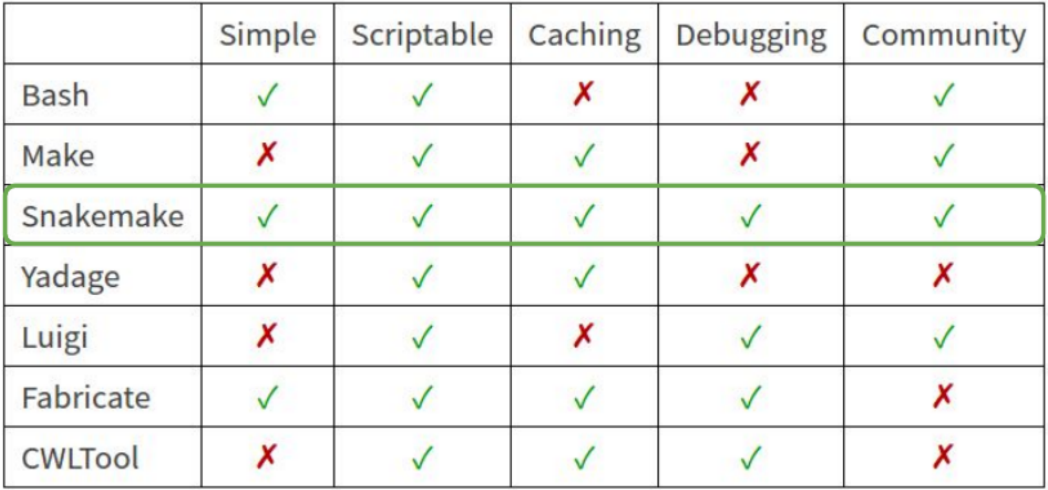
ntuples provide natural interface

Preserving = Re-running

Tools:Git&GitLab

- https://git-scm.com/
- http://cepcgit.ihep.ac.cn/
- Git submodule&subtree
- GitLab Continuous Integration(GitLab CI)
- GitLab Container Registry

Tools:Pipeline tools



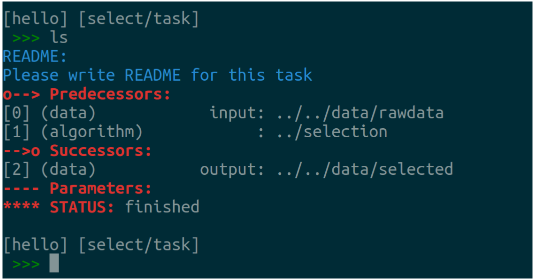
	Simple	Scriptable	Caching	Debugging	Community
Bash	✓	✓	✗	✗	✓
Make	✗	✓	✓	✗	✓
Snakemake	✓	✓	✓	✓	✓
Yadage	✗	✓	✓	✗	✗
Luigi	✗	✓	✗	✓	✓
Fabricate	✓	✓	✓	✓	✗
CWLTool	✗	✓	✓	✓	✗

Tools:Docker(containerized analysis)

- Highly recommend
- https://www.docker.com/
- Docker is the tool for containerized analysis.
- The developers use Docker to eliminate "Work on my machine" problems when collaborating on code with co-workers.
- Container:using containers, everything required to make a piece of software run is packaged into isolated containers.
- Always run the same, regardless of where it's deployed.

Tools:Chern

- A quality of life tool
- http://chern.readthedocs.io/en/latest/



```
[hello] [select/task]
 >>> ls
README:
Please write README for this task
o--> Predecessors:
[0] (data)             input: ../../data/rawdata
[1] (algorithm)             : ../selection
-->o Successors:
[2] (data)            output: ../../data/selected
---- Parameters:
**** STATUS: finished

[hello] [select/task]
 >>> 
```

Tools:REusable ANAlyses(CERN Analysis Preservation)

- REANA is a system that permits to instantiate research data analysis on the cloud. It uses container-based technologies and was born to target the use case of particle physics analyses in LHC collaborations.
- Instantiate workflows on the cloud.
- Manages job queues.
- Manages computing cloud resources.
- Support for OpenStack, Magnum, Kubernetes, EOS, Docker technologies.

Draft suggestions for CEPC analysis(minimal)

- Repository
  - complete analysis code on gitlab
  - accessible to the collaboration
- Analysis pipeline
  - Full instructions how to run the analysis
- Runtime environment
  - Instruction of how to set up environment
- Data storage
  - all input data on somewhere
  - readable by collaboration

Draft suggestions for CEPC analysis:Analysis respository

- Goals:
  - Preserve analysis tools and logic
  - Facilitate collaboration
  - Enable reuse of tools
- Recommendation:
  - Complete analysis code on gitlab
  - Fork&merge workflow
  - Modularize the analysis
  - Use separate repo for results and ANA

Draft suggestions for CEPC analysis:Modularizing projects

- might split responsibilities for different parts of the analysis
- tools can be shared between several analysis

Recommendation:

- One master repo
- include modules into the master
    - git submodule
    - git subtree

http://winstonkotzan.com/blog/2016/09/26/git-submodule-vs-subtree.html

Draft suggestions for CEPC analysis:Runtime Environment

- Use container
- Dockerfile kept in analysis repository
- More...

Draft suggestions for CEPC analysis:Input data

- Generator&Mokka data? –> "Official" production.
- Accessible to the whole group with documents to reduce the reuse of CPU time and Disks.
- Marlin data&ntuples (intermediate data)? solution?

Draft suggestions for CEPC analysis:Software

- Mokka&Marlin and etc on gitlab.
- fork&merge, forbid to use untracked processor.
- Share the tools and make them better together.
- Present not only results on meeting but also tools.

Conclusion & What to do?

- Conclusion
  - Analysis preservation will make the life better.
  - Not a lot effort, just try to use the new tools.
- What to do?
  - More details: https://goo.gl/ngAzhn
  - A finished analysis as demo.
  - More discussion now and via email.

Thanks