Results from Large-scale HPC deployment of Scalable CyberInfrastructure for Artificial Intelligence and Likelihood Free Inference (SCAILFIN)

Kenyi Hurtado, representing the SCAILFIN team

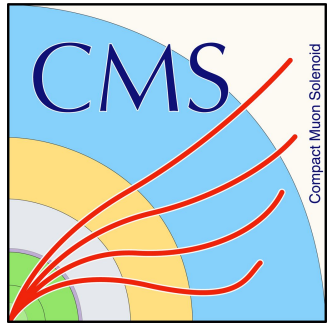
1

Mike Hildreth: mhildret@nd.edu; Scott Hampton: shampton@nd.edu ; Kenyi Hurtado: khurtado@nd.edu

Motivation

Major Multi-User Research Facilities involve the comparison of data collected from experiments with “synthetic" data, produced from computationally-intensive simulations.

Comparisons of experimental data and predictions from simulations are abstractions of the specific data analysis techniques developed by the respective communities over several decades. E.g.:



Motivation

Many of these data analysis tasks are often conducted manually or through *ad hoc scripts* that might not be well maintained, making reproducibility and reusability difficult. Many of these tasks do have a well-defined workflow that make automation possible, though.

REANA was created (in collaboration with DASPOS, DIANA and CERN) to address the reproducibility and reusability of the analysis pipeline.

Reproducible research data analysis platform

Motivation

In parallel:

Interest in leveraging Machine Learning (ML) and Artificial Intelligence (AI) techniques, to enhance the analysis of data from these facilities.

In particular, its application with emergent **Likelihood-Free Inference (LFI) techniques** when the predictions for the data are implicitly defined by the simulation, often leading to an intractable likelihood function. This can apply to analysis of data from LHC, LIGO, etc, but such Likelihood-Free algorithms have so far been **implemented mostly on individual machines and in ad hoc scripts because the training workflows are very complicated.**

Introduction

SCAILFIN: Scalable CyberInfrastructure for Artificial Intelligence and Likelihood Free Inference

The SCAILFIN project aims to deploy artificial intelligence and likelihood-free inference techniques and software using scalable cyberinfrastructure (CI) that is developed to be integrated into existing CI elements, such as the **REANA** system, to work on **HPC facilities**.

PI's: Mark Neubauer, Dan Katz

Kyle Cranmer, Heiko Mueller

Mike Hildreth

Introduction

REANA

Madminer / ML / AI

Scalability on HPC resources

Funding by the National Science Foundatioin:
NSF OAC-1841456
NSF OAC-1841471
NSF OAC-1841448

Introduction

NYU

REANA

Madminer / M

Today's focus: Extending the REANA platform to allow remote **submission** of workflows to **HPC facilities**

Scalability on HPC resources

Funding by the National Science Foundatioin:
NSF OAC-1841456
NSF OAC-1841471
NSF OAC-1841448

reana Reproducible Research Data Analysis Platform

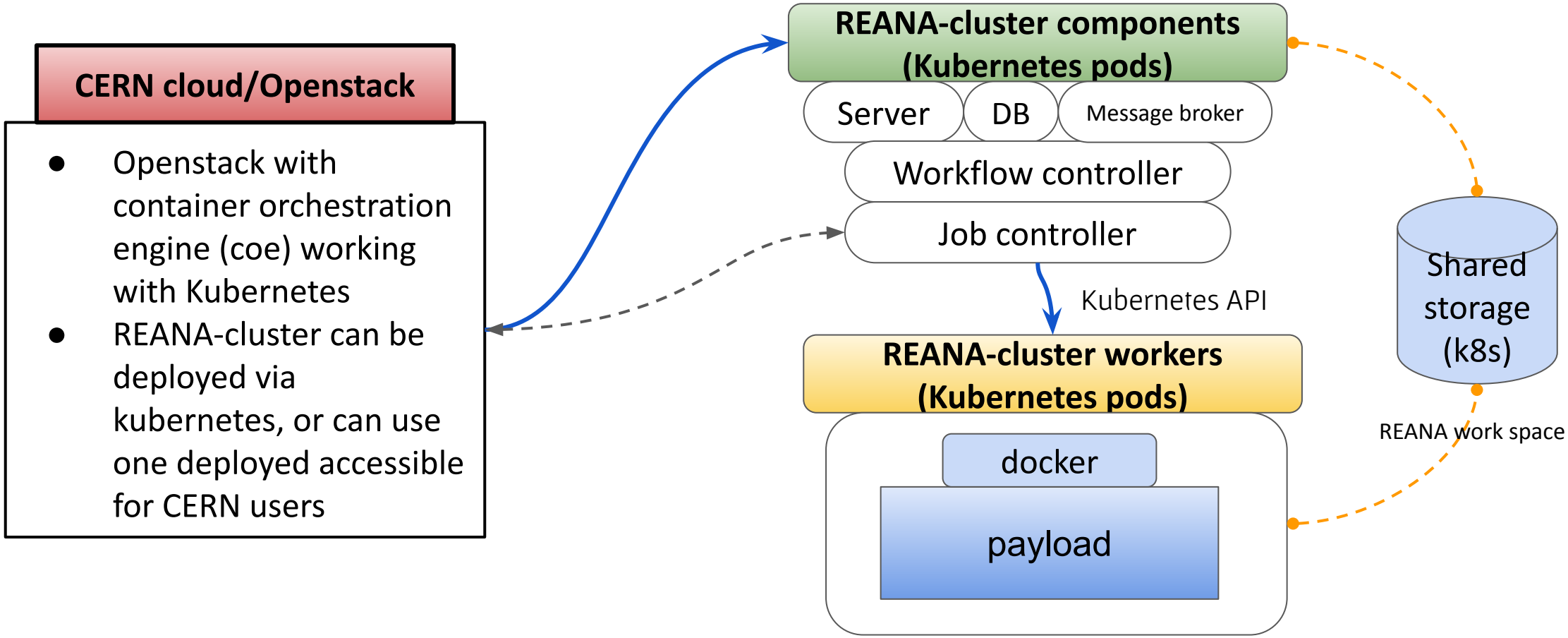
Components

- Two major components each consisting of many sub-components
  - **reana-client**: User facing component.
    - Accepts workflows and and is used as interface to entire REANA system (for user).
  - **reana-cluster**: Workhorse.
    - Consists of many small pieces which handle workflows, dish out jobs, coordinates results, can be thought of as the job scheduler. Jobs are scheduled via Kubernetes.

REANA cluster / Workforce Infrastructure
Standard kubernetes deployment

CERN cloud/Openstack

- Openstack with container orchestration engine (coe) working with Kubernetes
- REANA-cluster can be deployed via kubernetes, or can use one deployed accessible for CERN users



Container technologies in HPC facilities

- HPC centers are no strangers to the user's need for containers nowadays.
  - Docker is not an option though (security reasons)
  - User-space container technologies preferred instead. E.g.:
    - Apptainer/Singularity: PSC, TACC, Comet, etc.
    - Shifter: NERSC, Blue Waters
    - Above options above have mechanisms to run/convert docker images

SCAILFIN Developments to make this work:

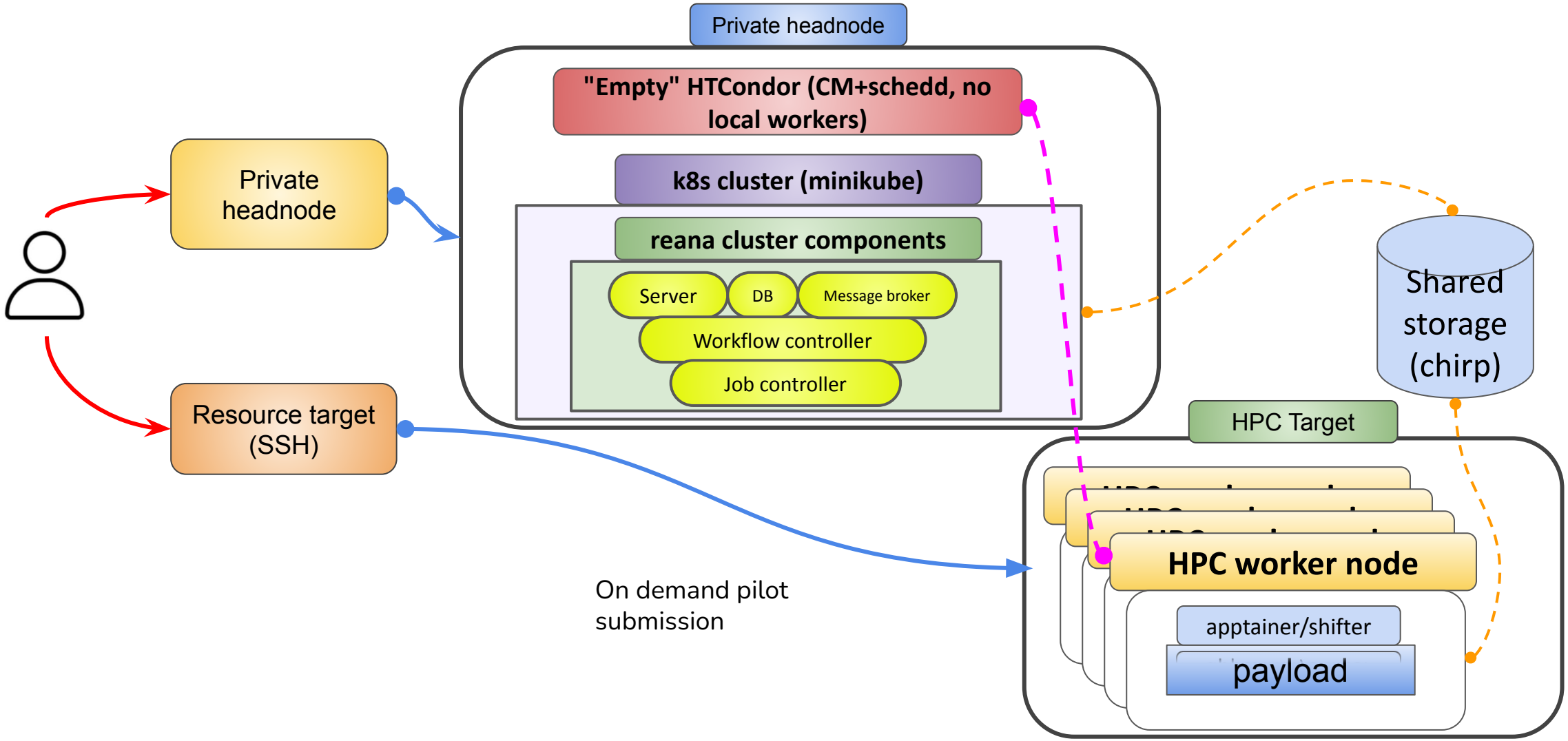
REANA Modifications:

- REANA requires some form of docker supporting container technology
  - Singularity and Shifter support finished.
- REANA expects to submit to a kubernetes cluster
  - Added support for VC3 specialized HTCondor submissions through a modified reana-job-controller and a job_wrapper for every workflow step.
  - The modified reana-job-controller submits each workflow step to a local condor scheduler
- Job Wrapper Auto-detection of container technology for workflow steps. (shifter, singularity)
- Relying on condor chirp to share files between the REANA cluster components and the workers

HPC submission

- Using VC3 pilot (HTCondor glidein) to launch HTCondor startds at the HPC target
- Pool password or token for authentication from workers to an empty private headnode condor schedd

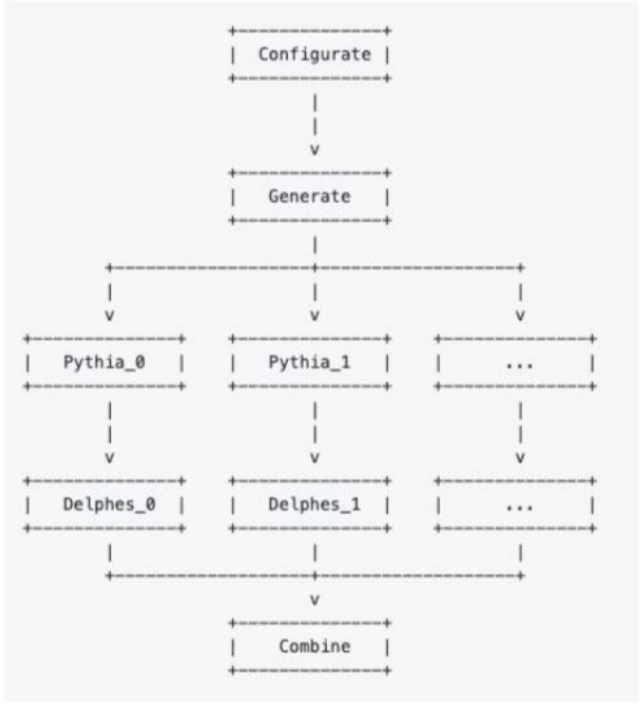
REANA cluster / Workforce Infrastructure
SCAILFIN deployment



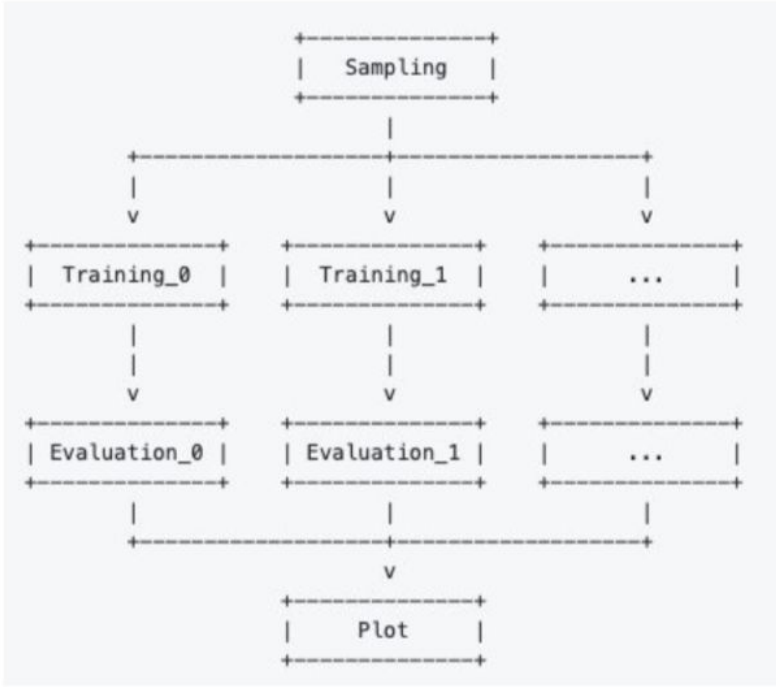
arXiv:1805.12244 - PRL
arXiv:1805.00013 - PRD
arXiv:1805.00020 - physics.aps.org/articles/v11/90

Madminer + REANA

Physics simulation:



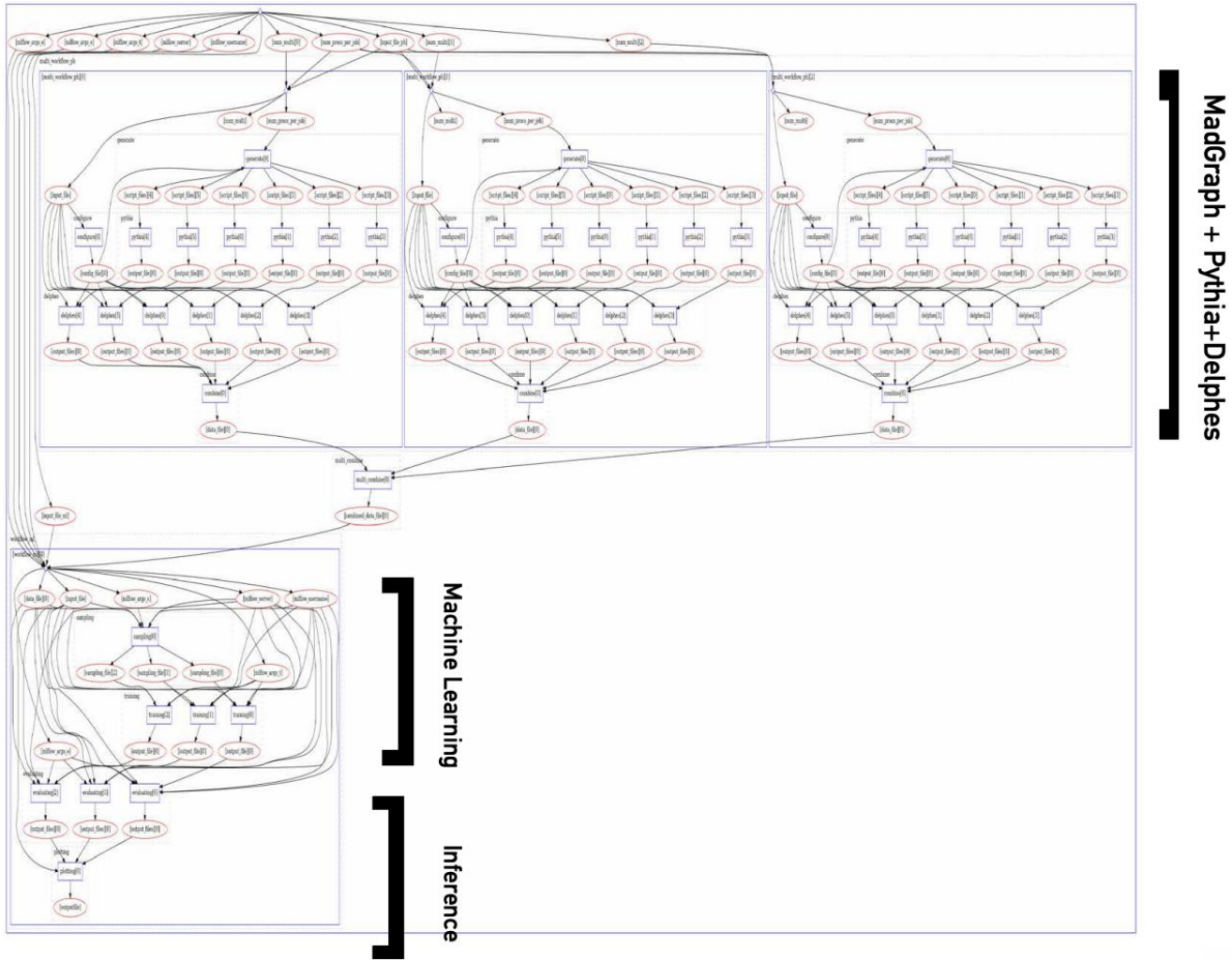
ML inference:



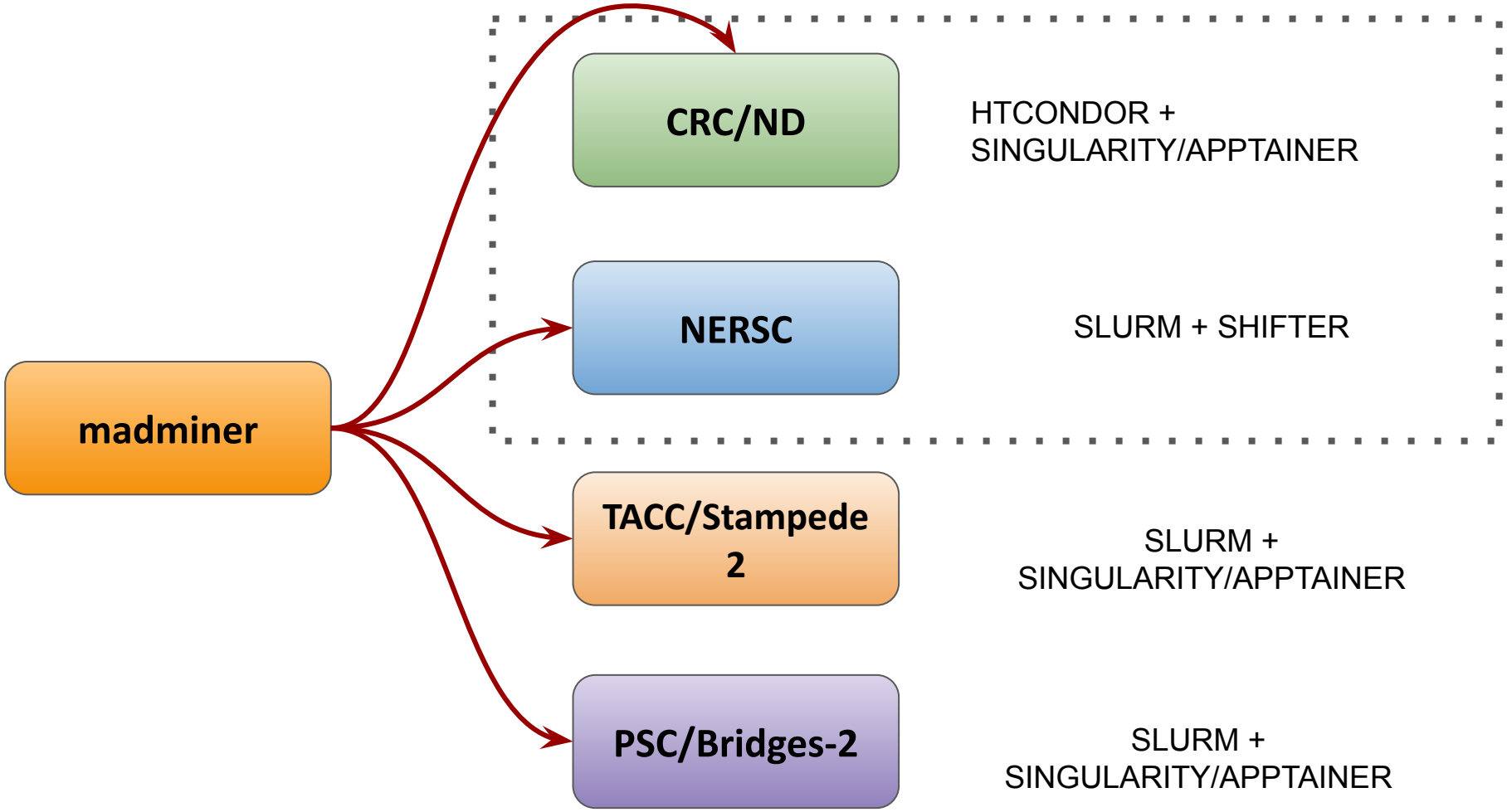
arXiv:1805.12244 - PRL
arXiv:1805.00013 - PRD
arXiv:1805.00020 - physics.aps.org/articles/v11/90

Madminer + REANA

To give you an idea of the workflow level structure and complexity



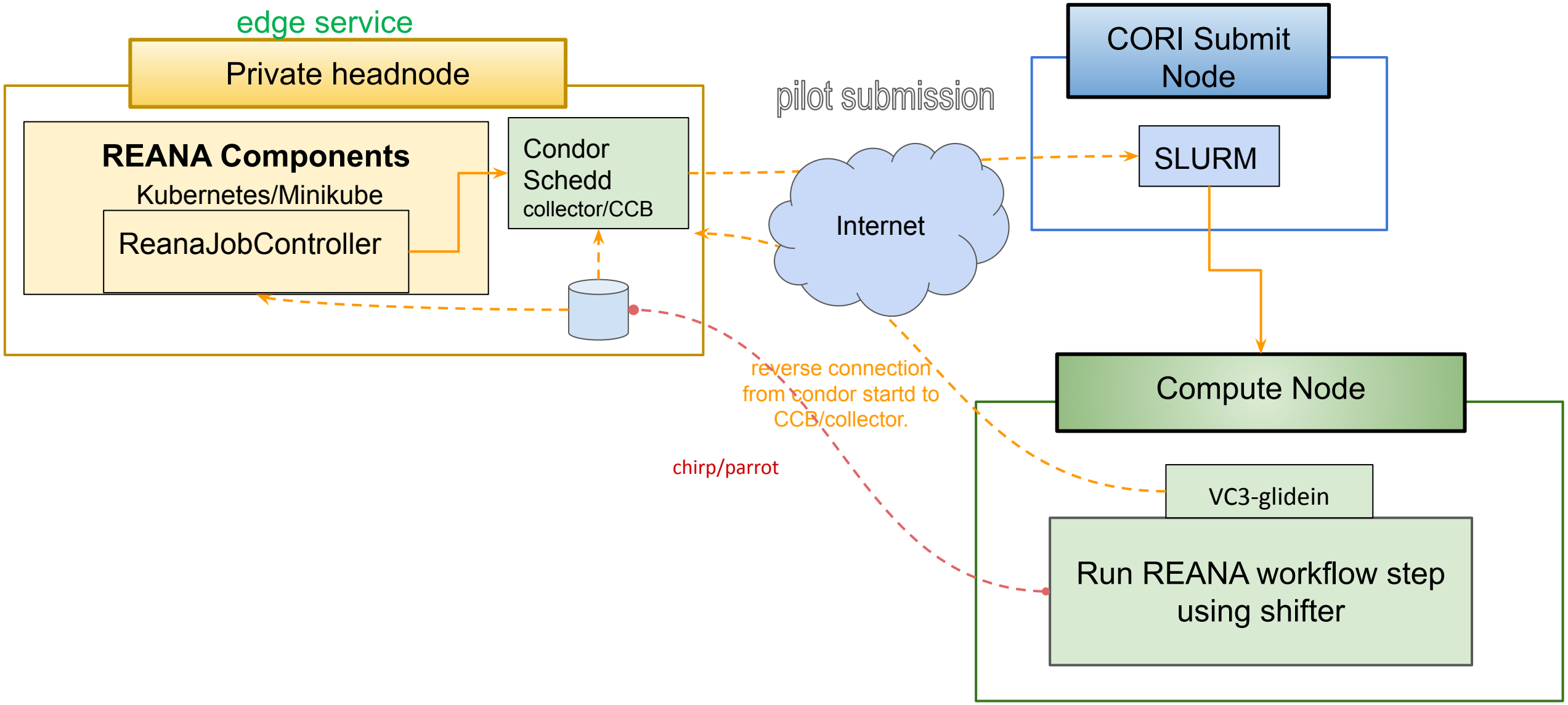
Running madminer



Infrastructure deployment example:

NERSC

SCAILFIN @NERSC



Scaling madminer from 6K to ~1M events

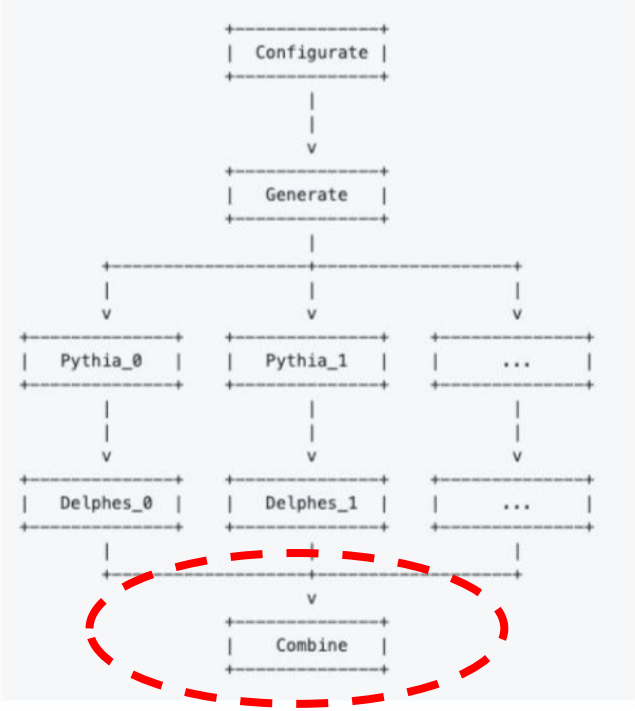
… so that we have enough statistics for this to be relevant for physics studies

Scaling madminer from 6K to ~1M events

Adapting Madminer

- Issue: Too many arguments!

Physics simulation:




```
combine_script.py </path/file1> </path/file2>
</path/file3> …
```

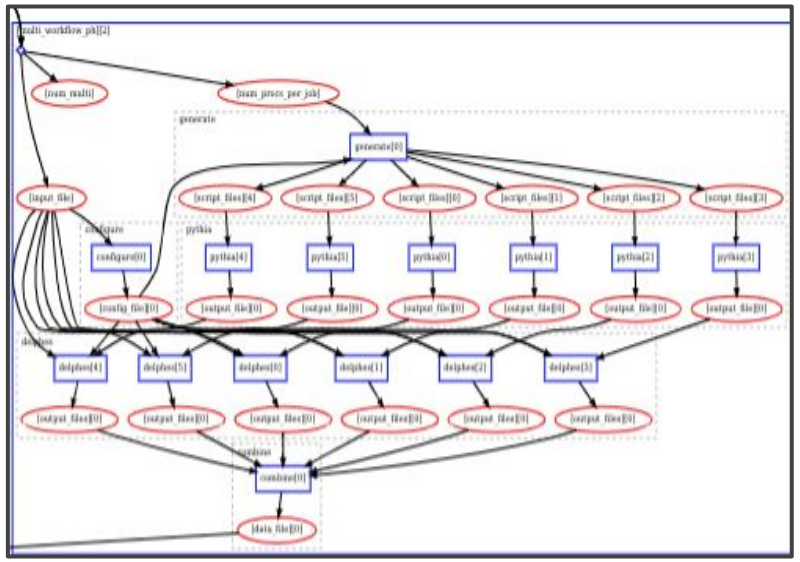
```
+ /usr/bin/singularity exec -B .//reana/use…
./tmp.K12VbTub4k: line 2: /usr/bin/singularity: Argument list too
long
[Error] Execution failed with error code: 126
```

It turns out, the number of arguments while combining all files would exceed the system's string length limit (ulimit settings)

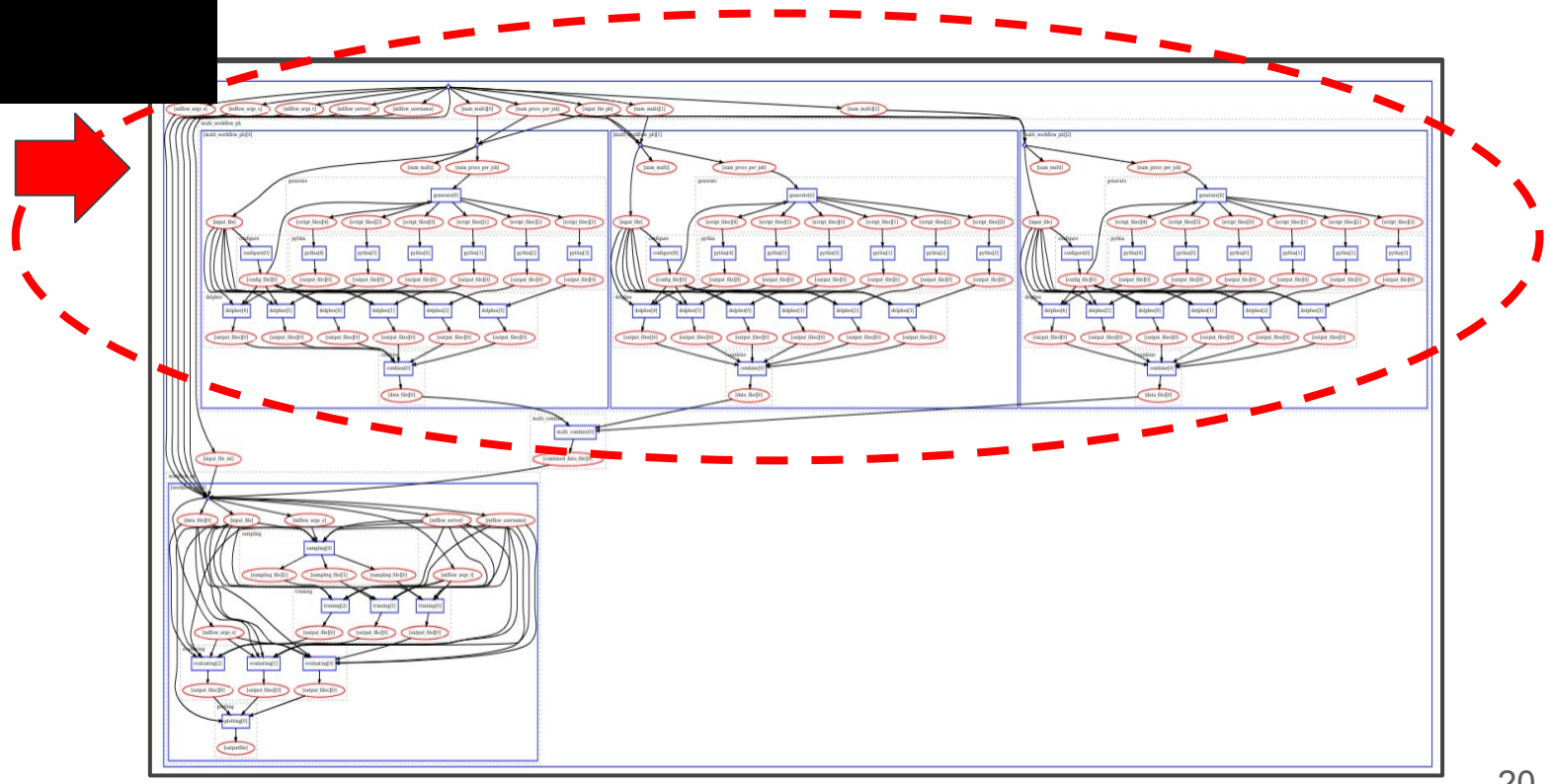
Scaling madminer from 6K to ~1M events

Adapting Madminer

```
+ /usr/bin/singularity exec -B .//reana/use…
./tmp.K12VbTub4k: line 2: /usr/bin/singularity: Argument
list too long
[Error] Execution failed with error code: 126
```



170 sub-workflows when generating the physics events (only 3 sub-workflows shown in the diagram)



Scaling madminer from 6K to ~1M events

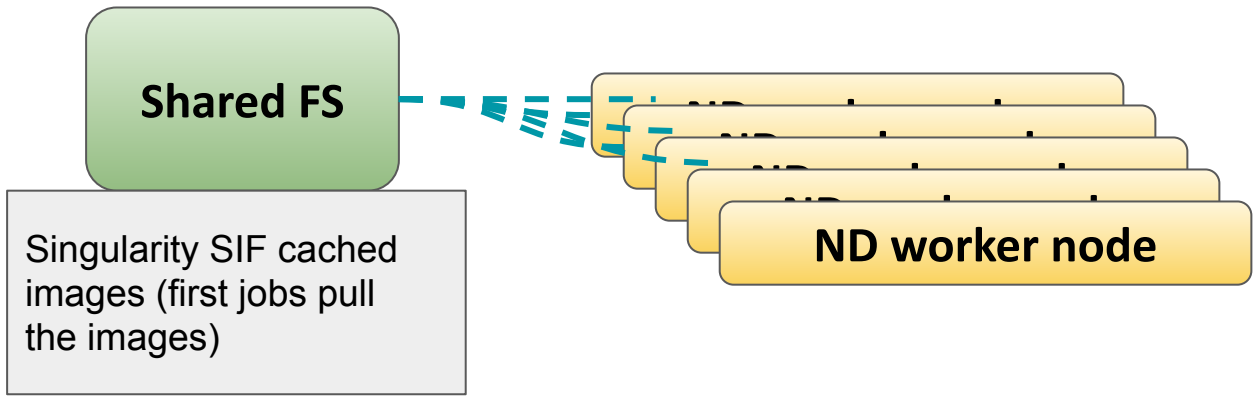
Running @ND

- Docker hub pull rate limit reached!

```
You have reached your pull rate limit. You may
increase the limit by authenticating and
upgrading:
https://www.docker.com/increase-rate-limits.
```

Solution: Caching

```
SINGULARITY_CACHEDIR=/path/to/shared/fs/working/area
```



Note we didn't see this issue at NERSC, since shifter already cached the images when converting them

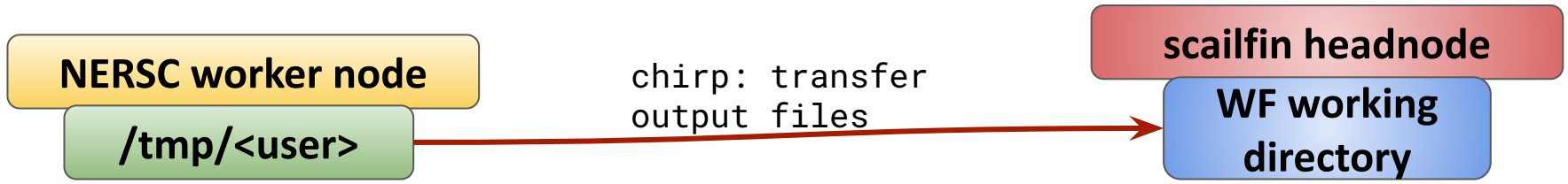
Scaling madminer from 6K to ~1M events
Running @NERSC

- HDF5 issues on LustreFS

```
-- Schedd: scailfin.crc.nd.edu : <>:9618?…
ID OWNER SUBMITTED RUN_TIME ST PRI SIZE CMD
482482.0 khurtado 2/16 20:11 0+08:01:05 R 0 0.0 delphes1…
```

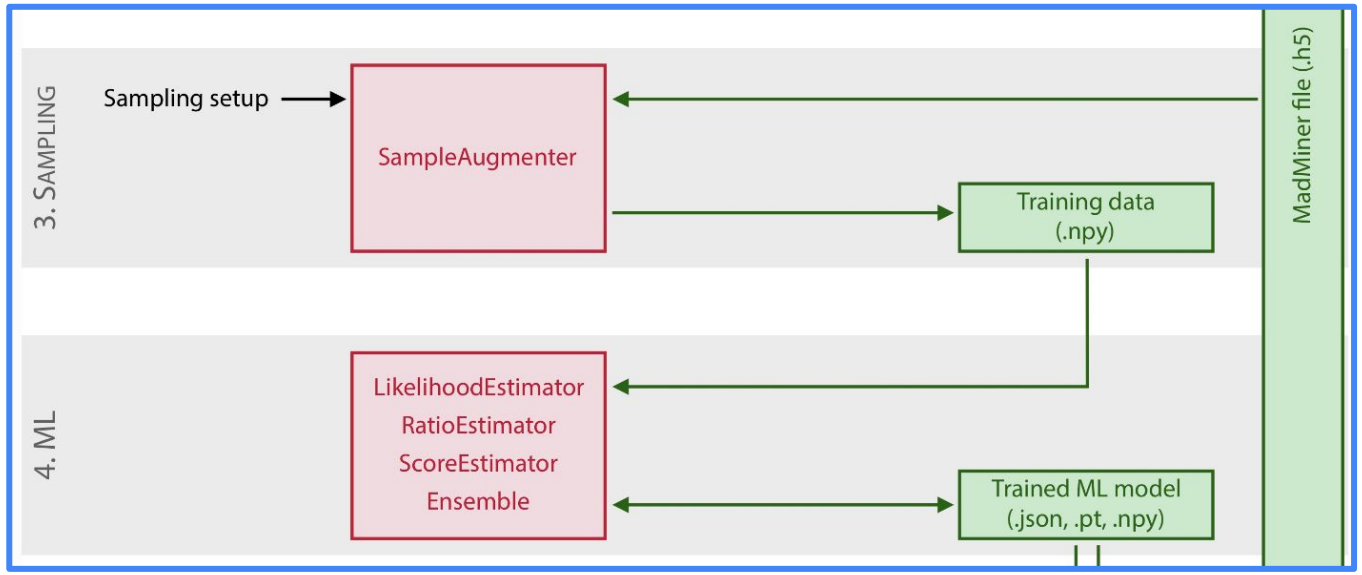
  - Jobs staying in running mode forever
    - Related to file locking issues with HDF5 files, which we use in the physics stage
    - `For some reason, HDF5_USE_FILE_LOCKING=FALSE`, suggested by NERSC was not enough to workaround the issue

Workaround: Run pilot on worker's temp area. Output files then are transferred to the submit node via condor_chirp, then cleanup the area.



Scaling madminer from 6K to ~1M events

Scaling the ML/Inference stage



Reference:
https://arxiv.org/pdf/1907.10621.pdf

Sampling stage parameters:

- **n_samples_train:** Number of events to be drawn from a training sample.
- **n_sampling_runs:** How many times to run the SampleAugmenter function

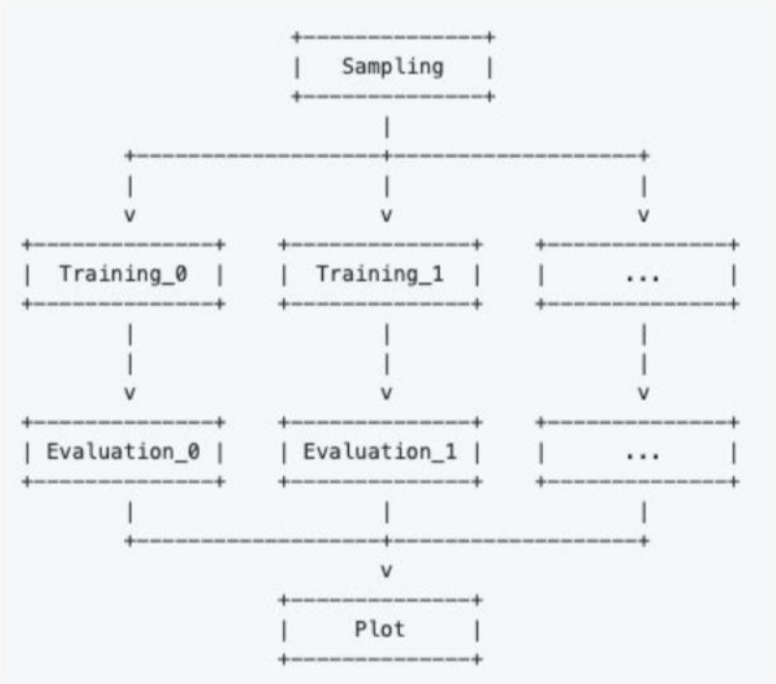
Sample unweighting and augmentation

```
score_estimators = ["alice", "alices", "sally"]
for method in score_estimators:
    for i in range(n_sampling_runs):
        SampleAugmenter[method](n_samples_train,
**other_parameters)
```

Scaling madminer from 6K to ~1M events

Scaling the ML/Inference stage

ML inference:



Number of jobs per step:

- Sampling: 1
- Training & Evaluation: n_score_methods * n_sampling_runs
- Plotting: 1

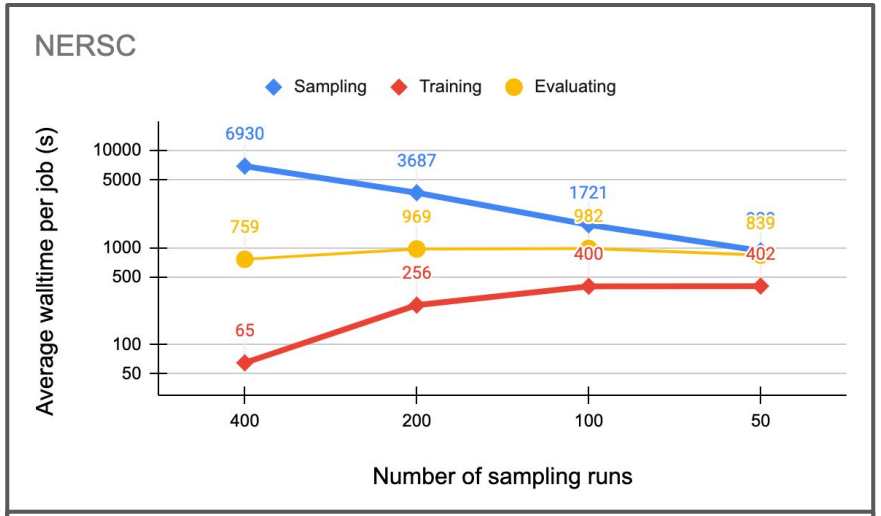
We are using 3 score methods in this test, so we can scale the number of jobs via **n_sampling_runs** and adjusting **n_samples_train** accordingly, to unweight&augment all the data.

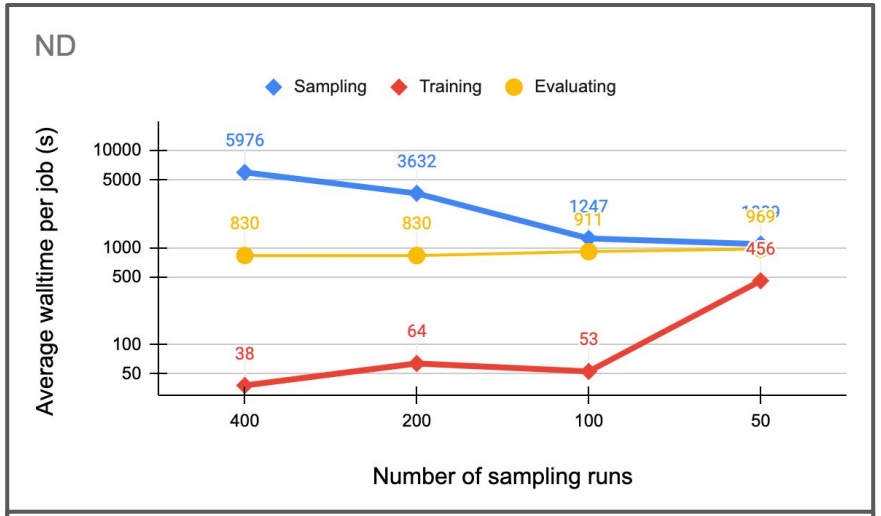
n_sampling_runs	400	200	100	50
n_sample_trains	2550	5100	10200	20400
Sample jobs	1	1	1	1
Training jobs	1200	600	300	150
Evaluating jobs	1200	600	300	150

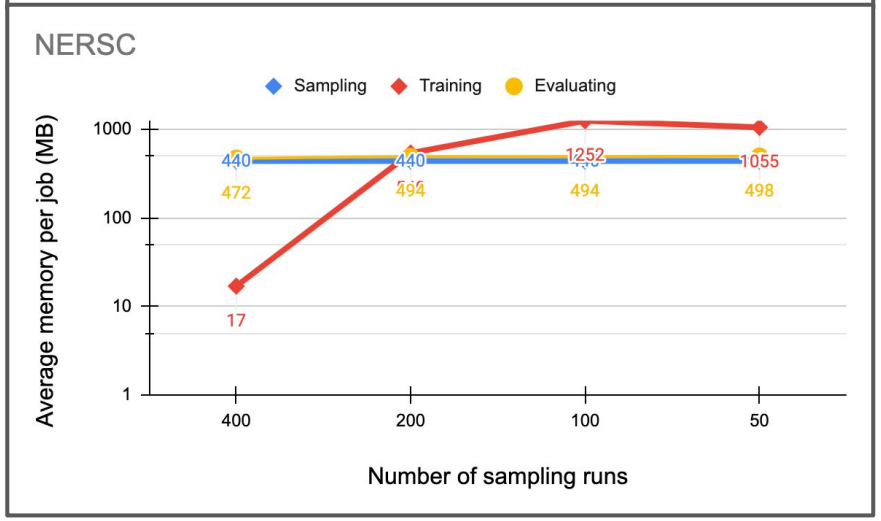
```
total_training_sample_events =  n_sampling_runs * n_samples_train
```

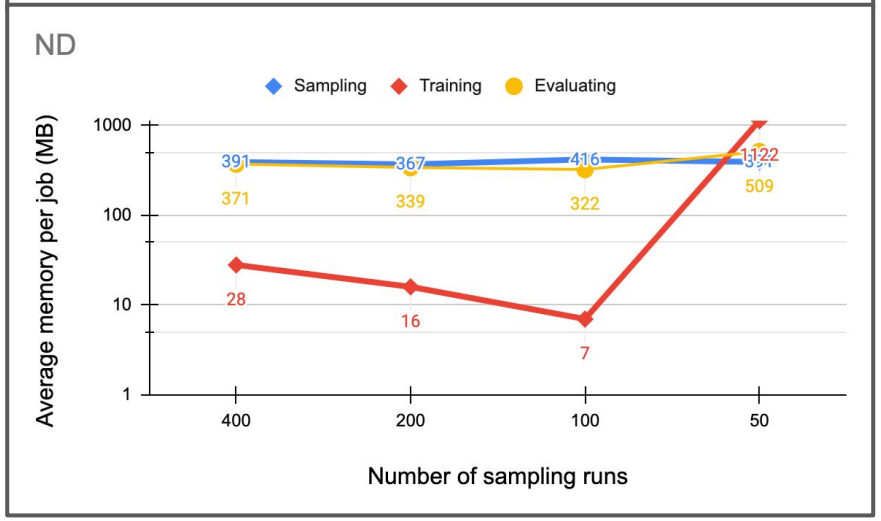
Scaling madminer from 6K to ~1M events

Average Walltime and Memory per job comparisons









- Sampling jobs scale fairly linearly, memory usage doesn't change
- No significant difference with evaluating jobs
- Training jobs average walltime and memory does overall get reduced with more sampling runs and less events job

Summary

- The SCAILFIN project allows the deployment of artificial intelligence and likelihood-free inference techniques for particle physics analyses. We have shown how the project uses scalable cyberinfrastructure (CI) that is developed to be integrated into existing CI elements, such as the REANA system, to work on HPC facilities.
- Successfully scaled the madminer workflow to an order of magnitude that becomes statistically relevant for physics studies using HPC resources

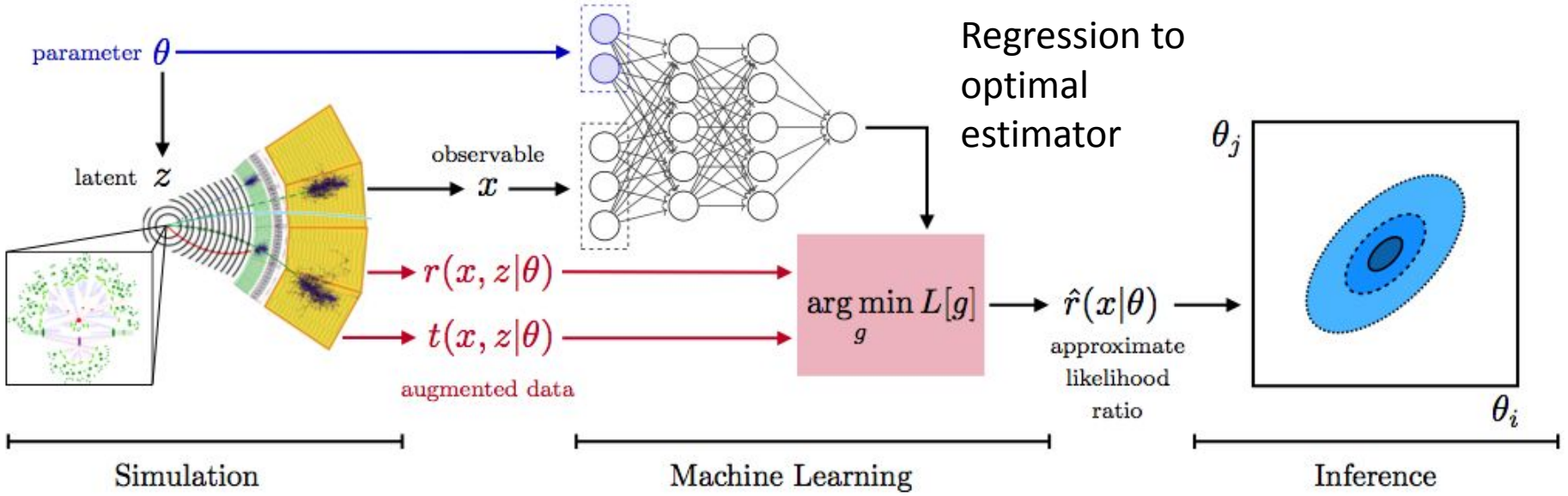
Thank you!

khurtado@nd.edu

Backup slides

arXiv:1805.12244 - PRL
arXiv:1805.00013 - PRD
arXiv:1805.00020 - physics.aps.org/articles/v11/90

Simulation-Based Likelihood Free Inference

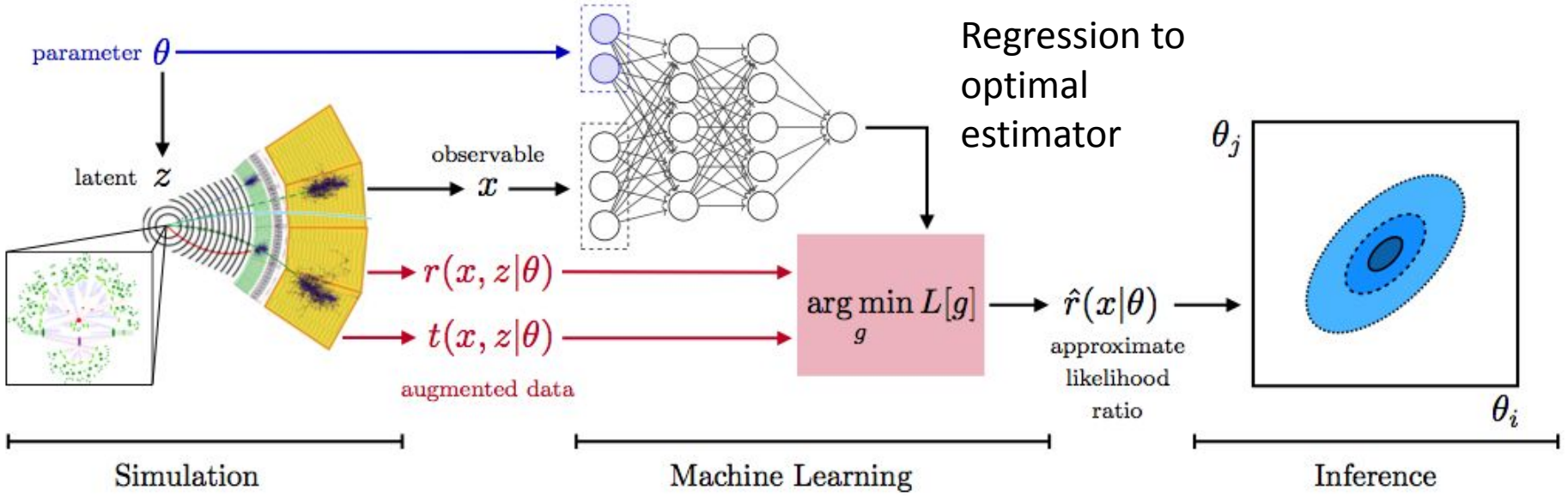


Estimation of optimal estimator lends itself to ML methods:

- Training data derived from simulations
- Can be guided by optimal sampling based on phase space density of generator, sensitivity to physics under study

arXiv:1805.12244 - PRL
arXiv:1805.00013 - PRD
arXiv:1805.00020 -
physics.aps.org/articles/v11/90

Simulation-Based Likelihood Free Inference



- ***Simulation-based inference***: we want to infer theoretical parameters using a simulator to describe predictions.
- ***MadMiner***: a tool that implements various simulation-based inference strategies for particle physics.

arXiv:1805.12244 - PRL, arXiv:1805.00013 - PRD, arXiv:1805.00020 - physics.aps.org/articles/v11/90

reana: Reproducible Research Data Analysis Platform

Features

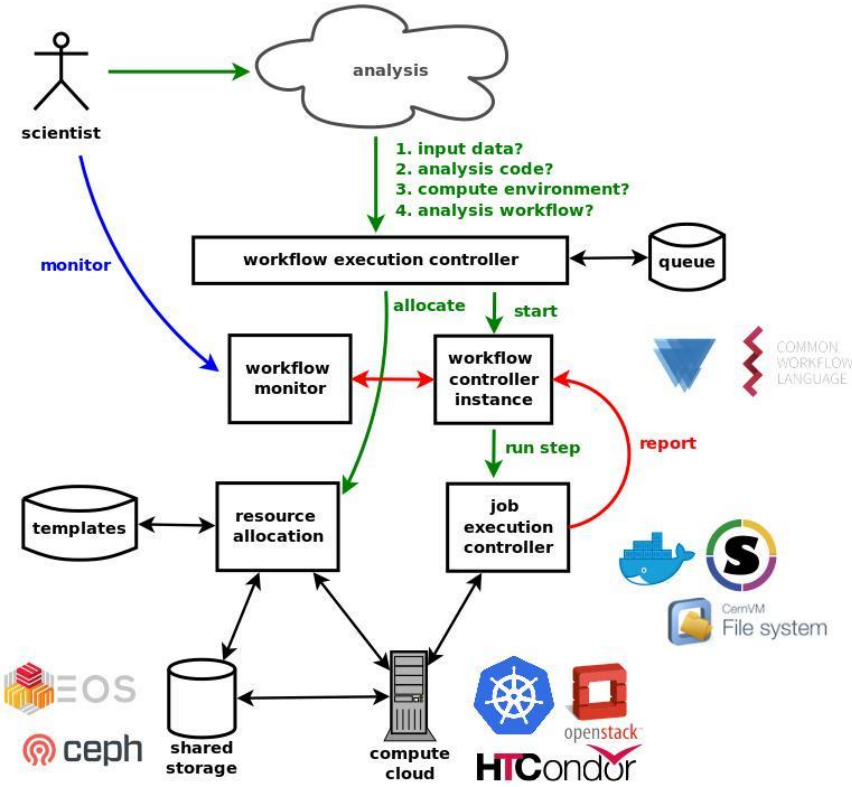
- Allows creation of tightly defined, container encapsulated workflows
- Built with commodity pieces
- Purpose is to allow complete reproducibility
- Sharing workflows is as easy as sharing a specification
  - (and inputs!)
- Different workflow engines supported. e.g.:
  - CWL (Common Workflow Language) : https://www.commonwl.org/
  - Yadage (YAML based adage): https://yadage.readthedocs.io

reana: Reproducible Research Data Analysis Platform

Process

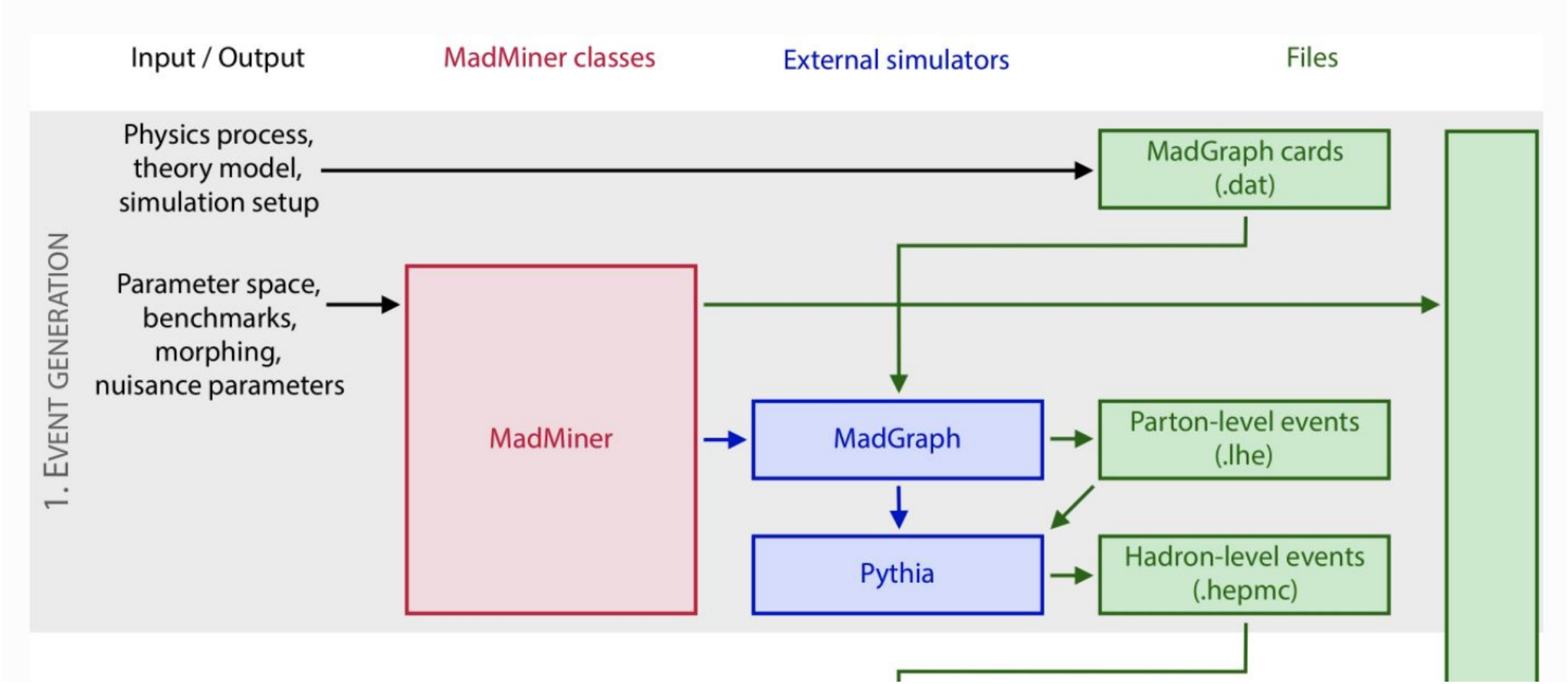
1. Create workflow specification (Yadage, CWL, Serial)
2. Upload workflow and inputs to REANA cloud
3. Start workflow
4. Download / pull down results
5. Share workflow specs with others

Architecture



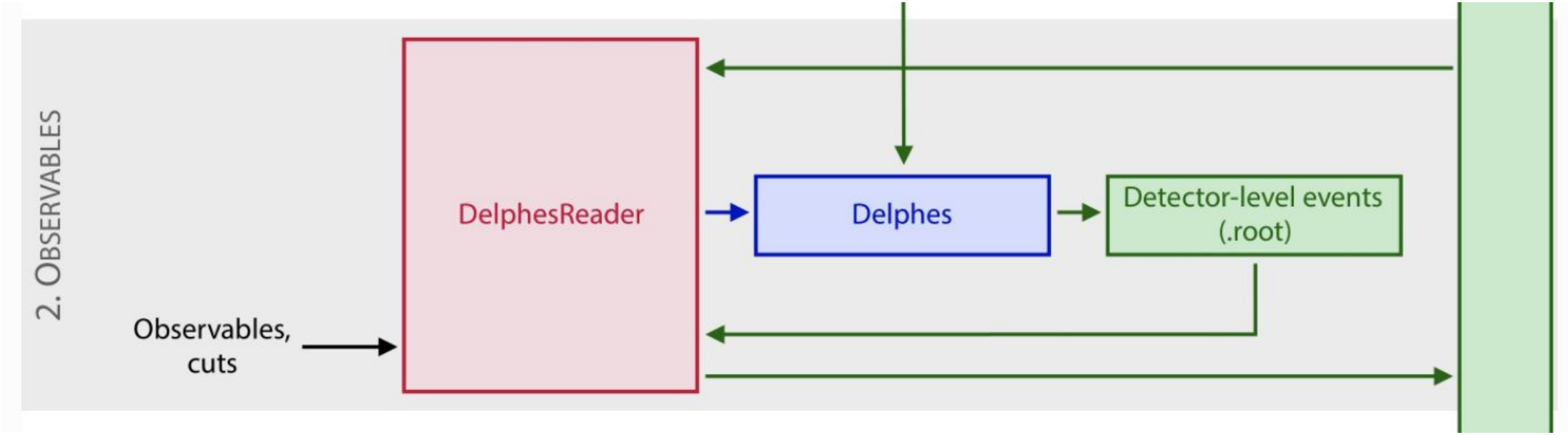
arXiv:1805.12244 - PRL
arXiv:1805.00013 - PRD
arXiv:1805.00020 - physics.aps.org/articles/v11/90

Simulation-Based Likelihood Free Inference



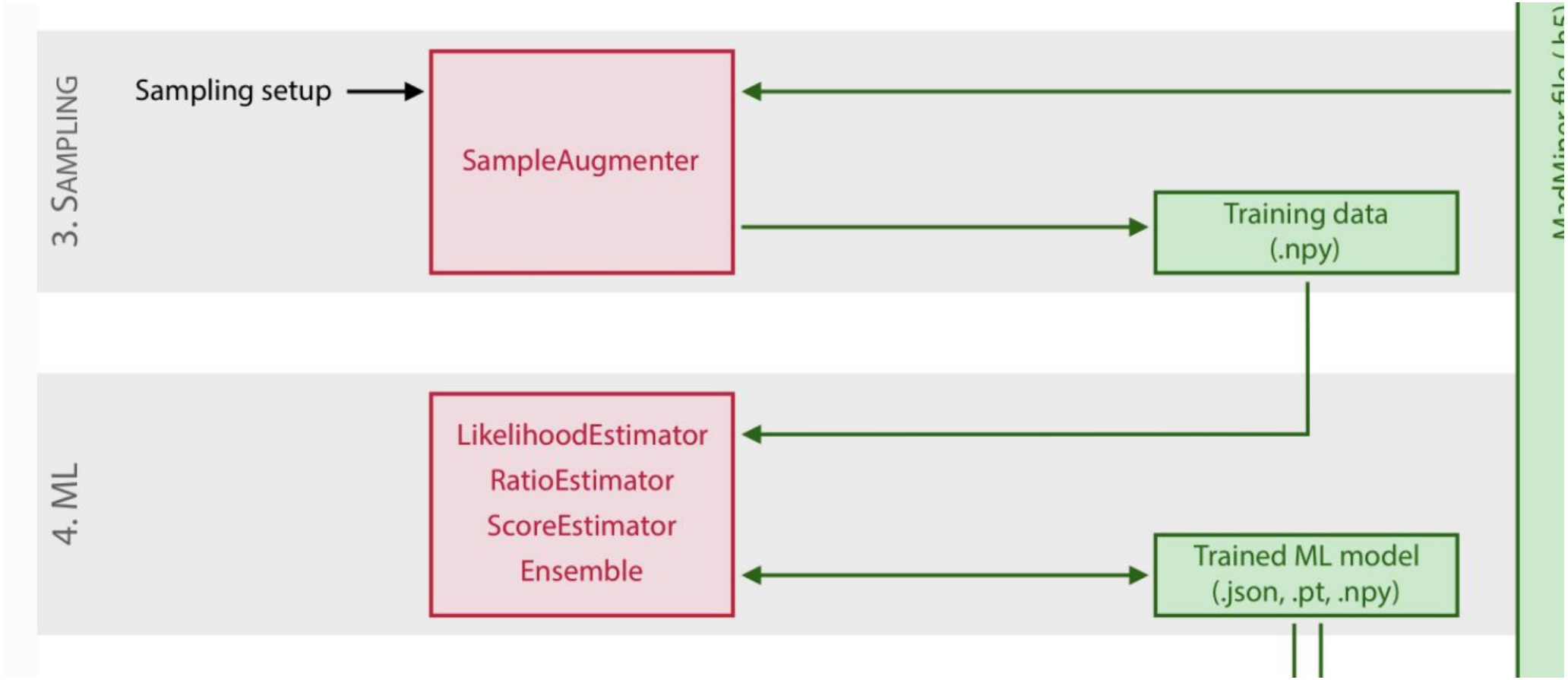
arXiv:1805.12244 - PRL
arXiv:1805.00013 - PRD
arXiv:1805.00020 - physics.aps.org/articles/v11/90

Simulation-Based Likelihood Free Inference



arXiv:1805.12244 - PRL
arXiv:1805.00013 - PRD
arXiv:1805.00020 - physics.aps.org/articles/v11/90

Simulation-Based Likelihood Free Inference

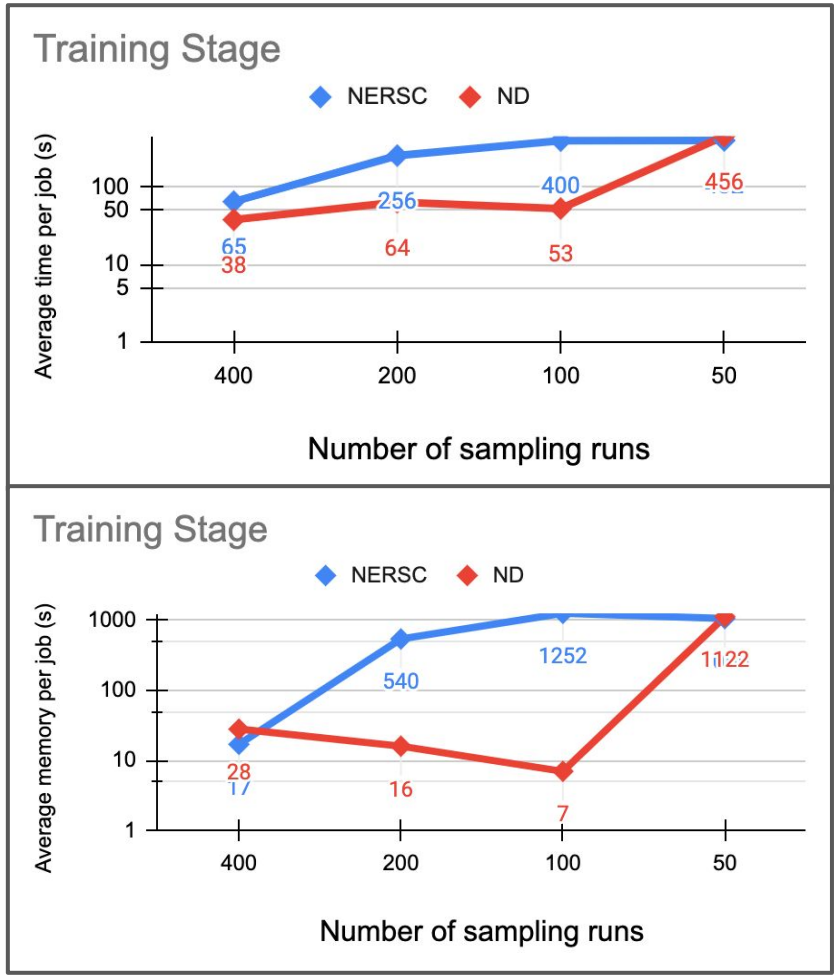


arXiv:1805.12244 - PRL
arXiv:1805.00013 - PRD
arXiv:1805.00020 - physics.aps.org/articles/v11/90

Simulation-Based Likelihood Free Inference

Scaling madminer from 6K to ~1M events

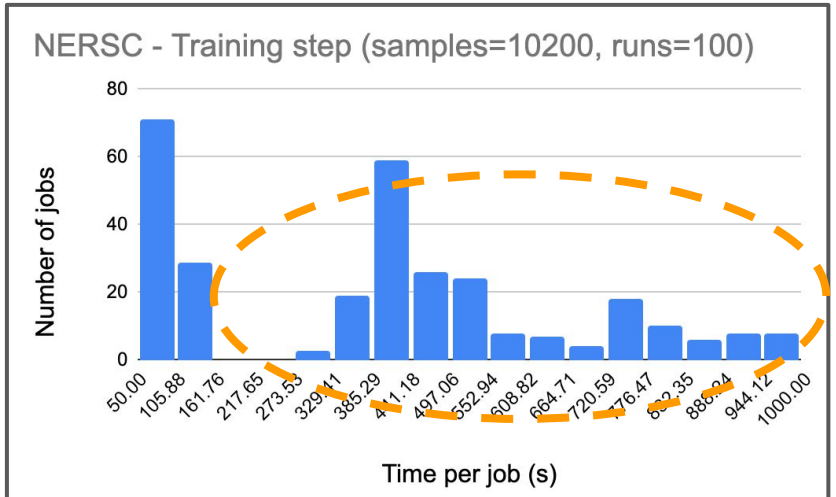
Average Walltime and Memory per job comparisons

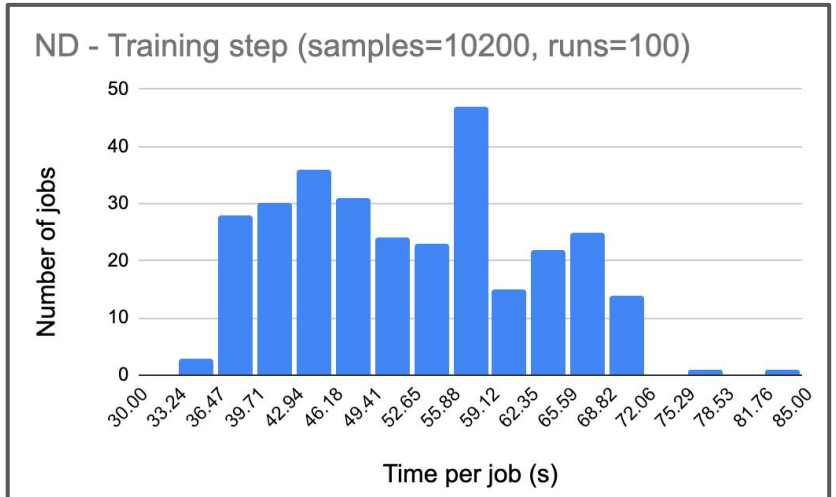


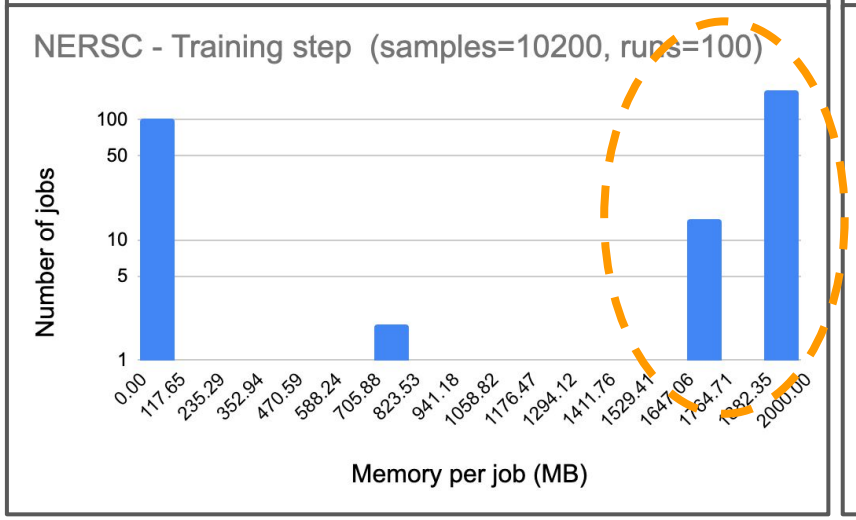
Differences noticed mainly in
n_sampling_runs = 100,200

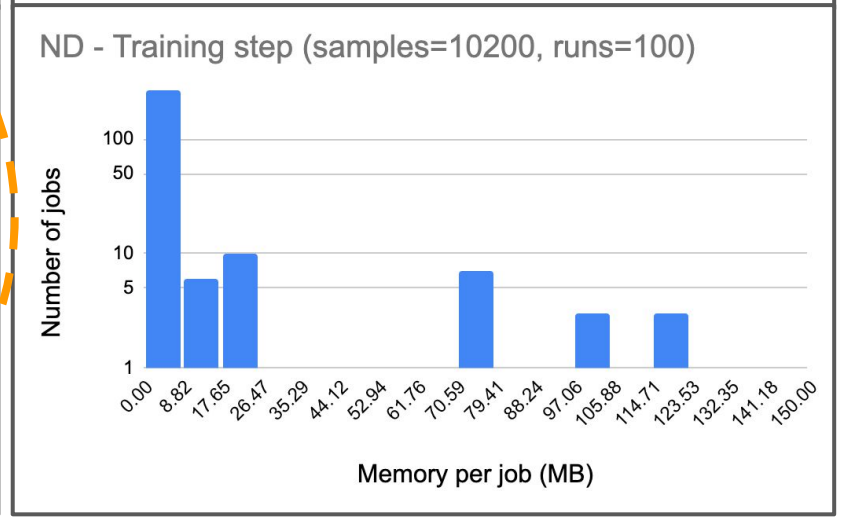
Scaling madminer from 6K to ~1M events

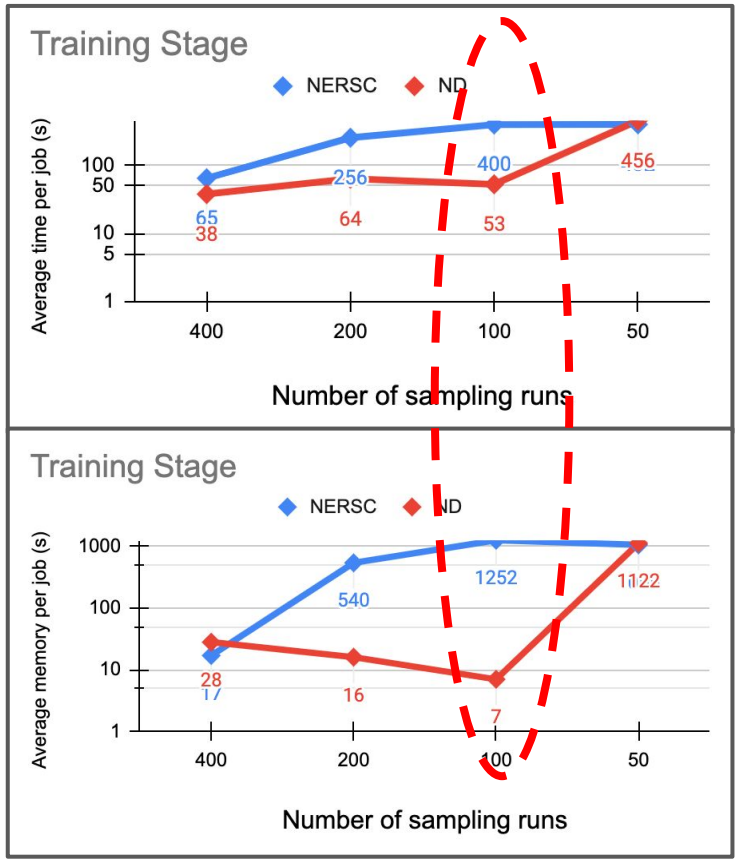
Average Walltime and Memory per job comparisons









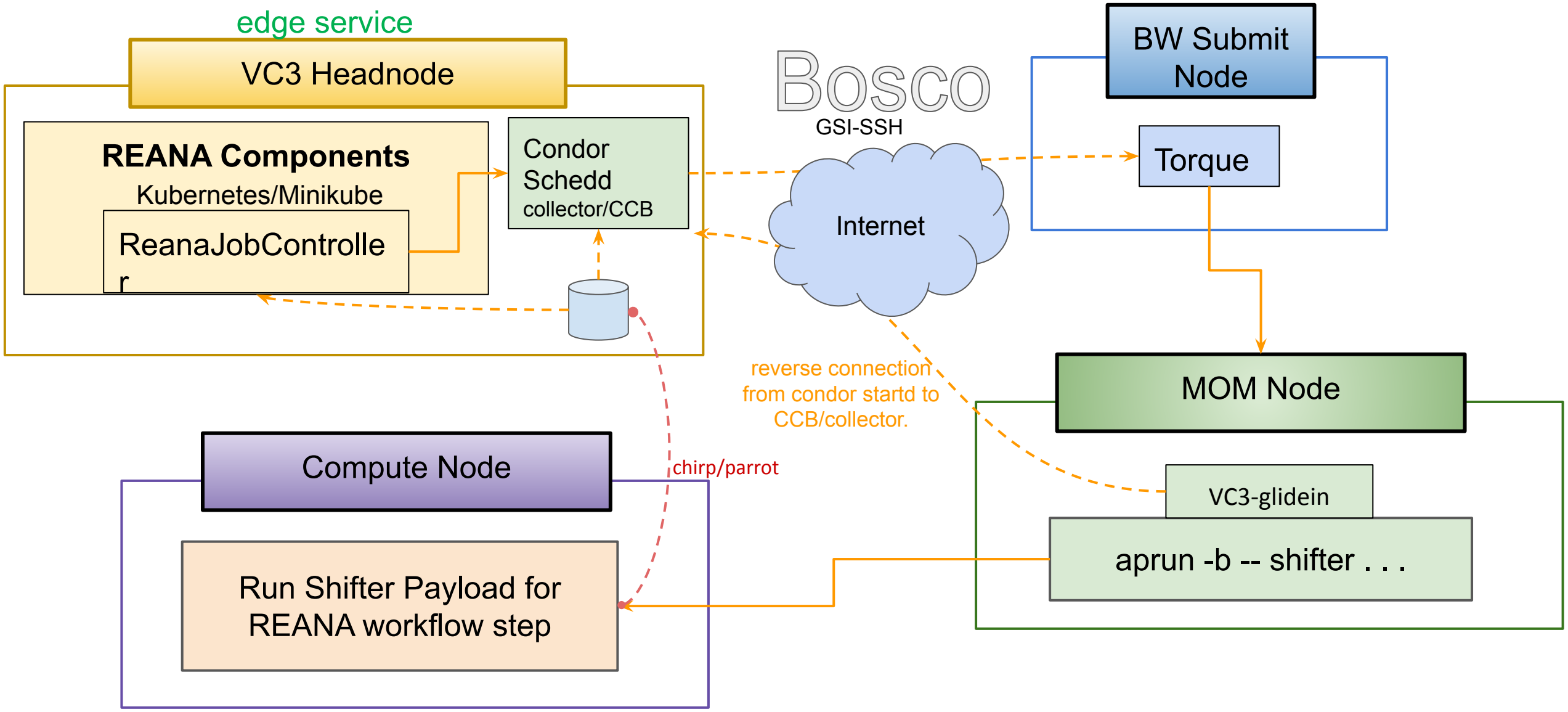


Notre Dame CCTools

- **Chirp:** Integrated with HTCondor. Adding "+WantIOProxy=True" creates a chirp server than can be used between the VC3-headnode and the workers. It also takes care of the authentication.

- **Parrot:** Static version available, runs on any x86 architecture. Can be used to interact with the chirp server created by HTCondor.
  - Note HTCondor has its own chirp client, but it doesn't e.g.: recursive copy directories.

SCAILFIN on Blue Waters



Blue Waters Tests...

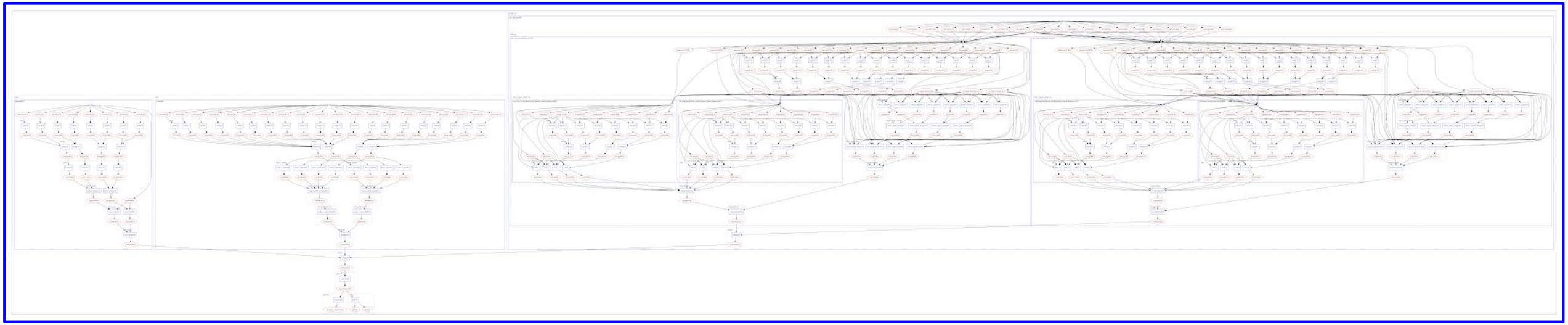
Successfully ran a complex physics test: BSM search

```
(reana) [khurtado@khurtado-reanabwvc3 condor]$ reana-client status
NAME        RUN_NUMBER   CREATED              STATUS     PROGRESS
demobsmv2   1            2019-06-20T14:24:40  finished   18767/65

(reana) [khurtado@khurtado-reanabwvc3 condor]$ reana-client ls | grep pdf
plot/prefit.pdf                                                    19405
2019-06-20T15:31:49
plot/postfit.pdf                                                   19431
2019-06-20T15:31:49

(reana) [khurtado@khurtado-reanabwvc3 condor]$ condor_history -constraint
'regexp("12610aee-f019-4888-85e5-e9c098d28bf8", Args)' -af:h ClusterId ExitCode LastRemoteHost
ClusterId ExitCode LastRemoteHost
97        0        slot1_1@31738@nid25424
96        0        slot1_1@13090@nid27638
95        0        slot1_1@13090@nid27638
94        0        slot1_1@13090@nid27638
93        0        slot1_1@13090@nid27638
```

Blue Waters Tests...

Successfully ran a complex physics test: BSM search

Here is the workflow diagram, to give you an idea of the complexity.



Blue Waters Tests...

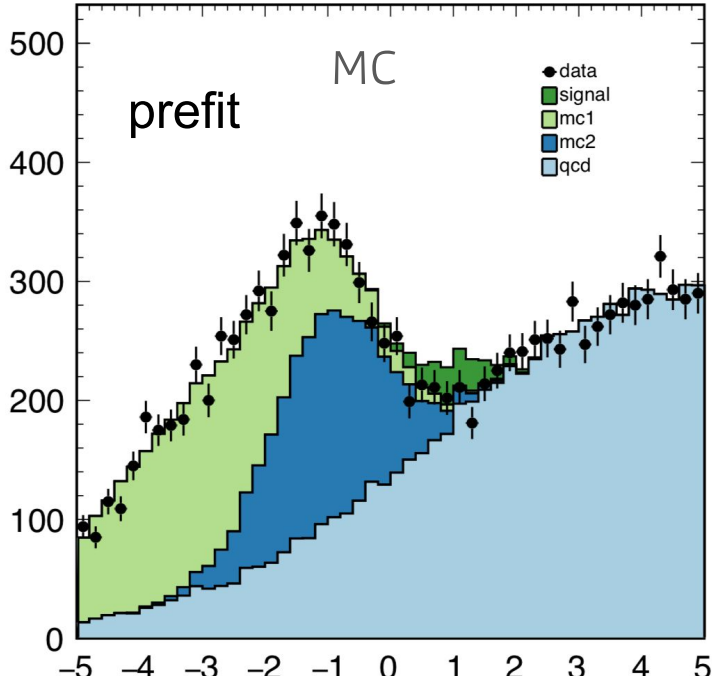
Once results are downloaded through the client, final plots look like this:

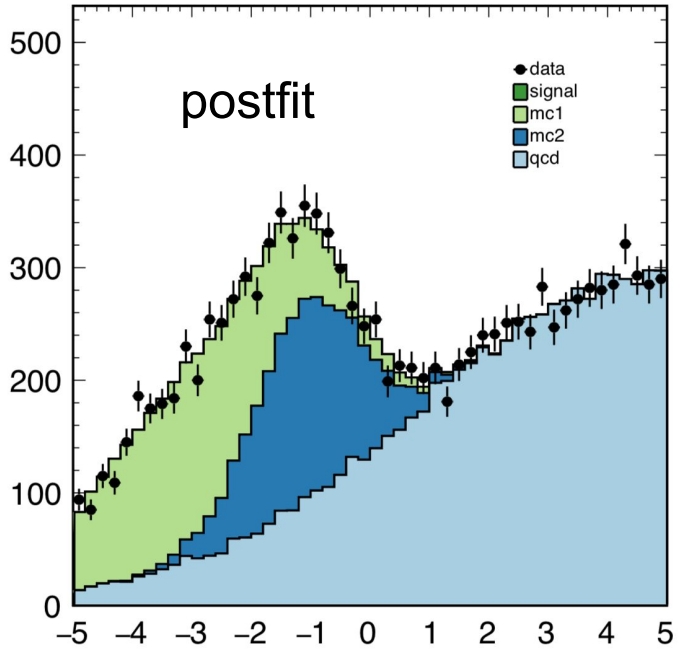
```
(reana) [khurtado@khurtado-reanabwvc3 condor]$ reana-client download plot/prefit.pdf
File plot/prefit.pdf downloaded to /home/khurtado/condor.
(reana) [khurtado@khurtado-reanabwvc3 condor]$ reana-client download plot/postfit.pdf
File plot/postfit.pdf downloaded to /home/khurtado/condor.
```

Data is generated/emulated according to Standard Model expectations.

After processing, a statistical model involving both signal and control regions is built and the model is fitted against the observed data.

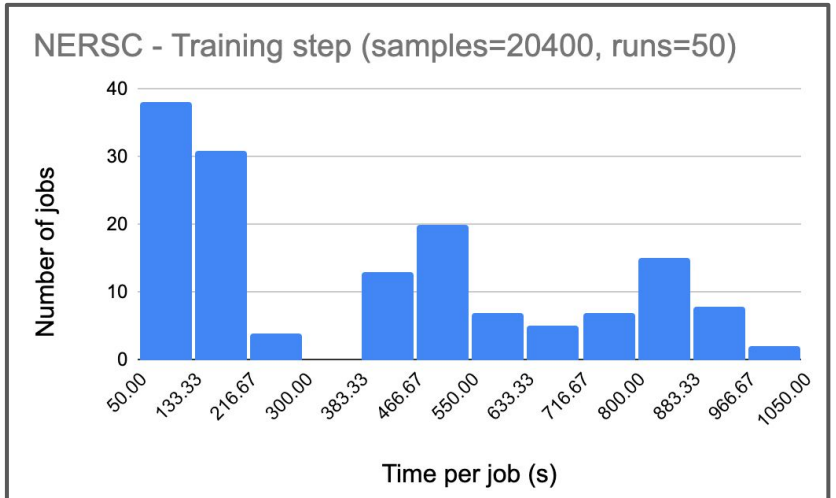
The signal sample is scaled down significantly to fit the data, which is expected since the data was emulated in accordance with a SM-only scenario

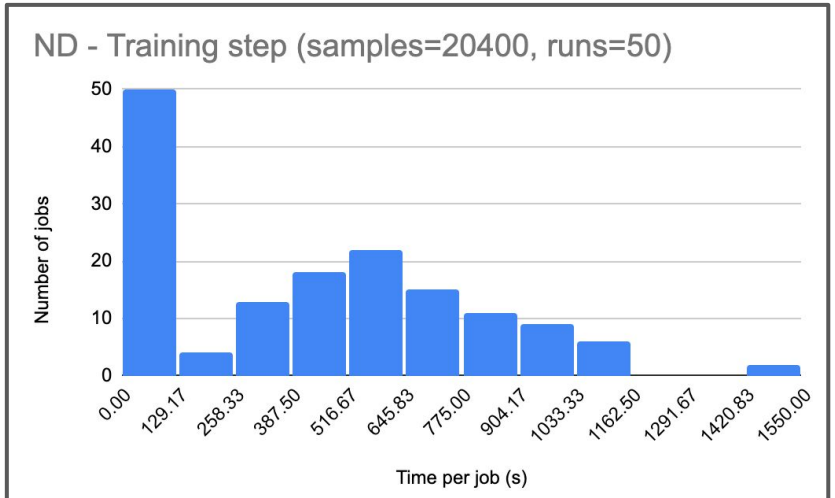


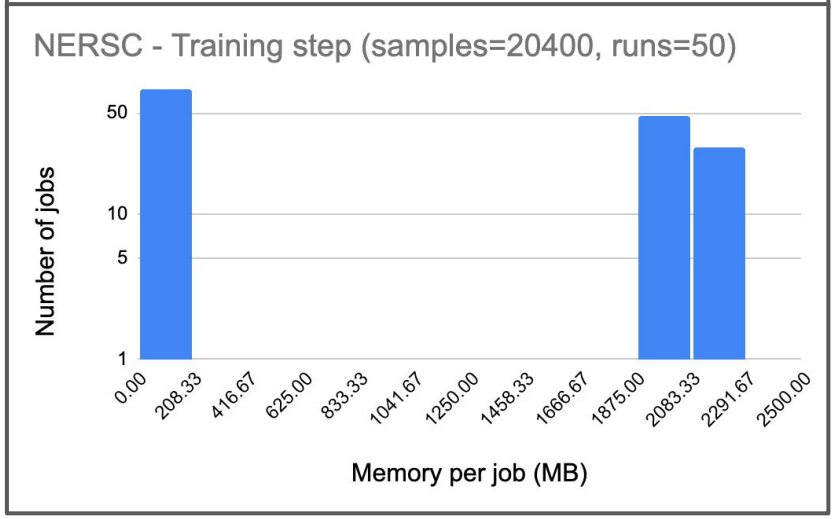


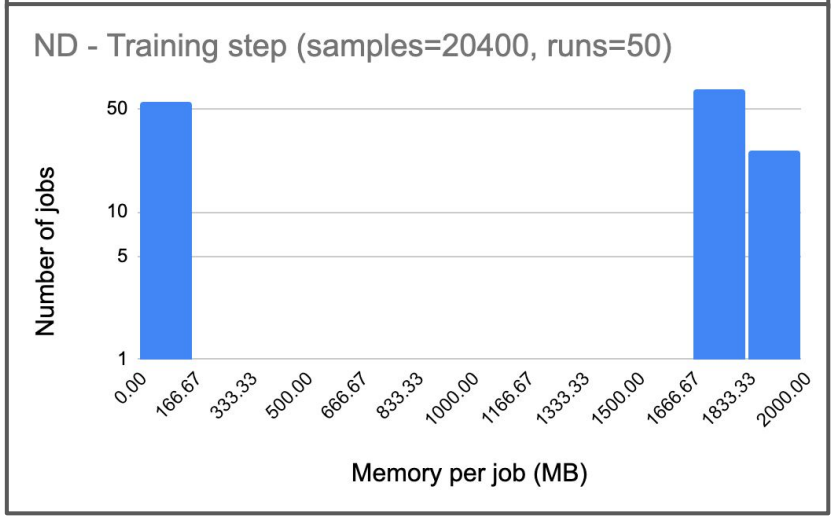
Scaling madminer from 6K to ~1M events

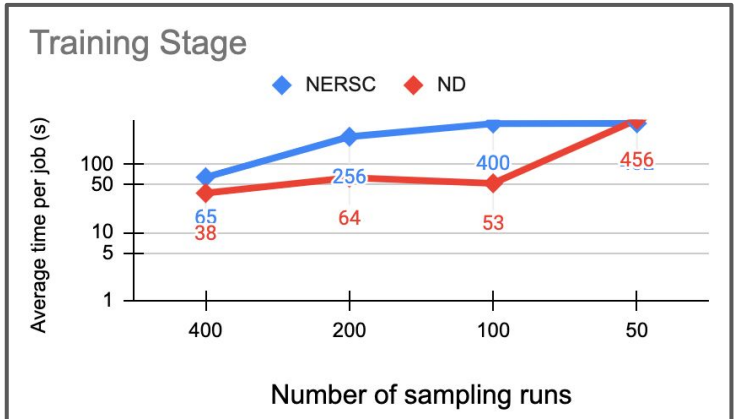
Average Walltime and Memory per job comparisons

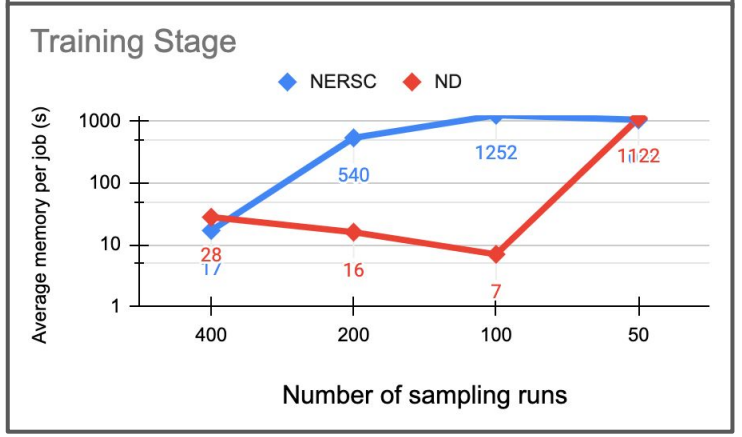












What's different about HPC facilities?

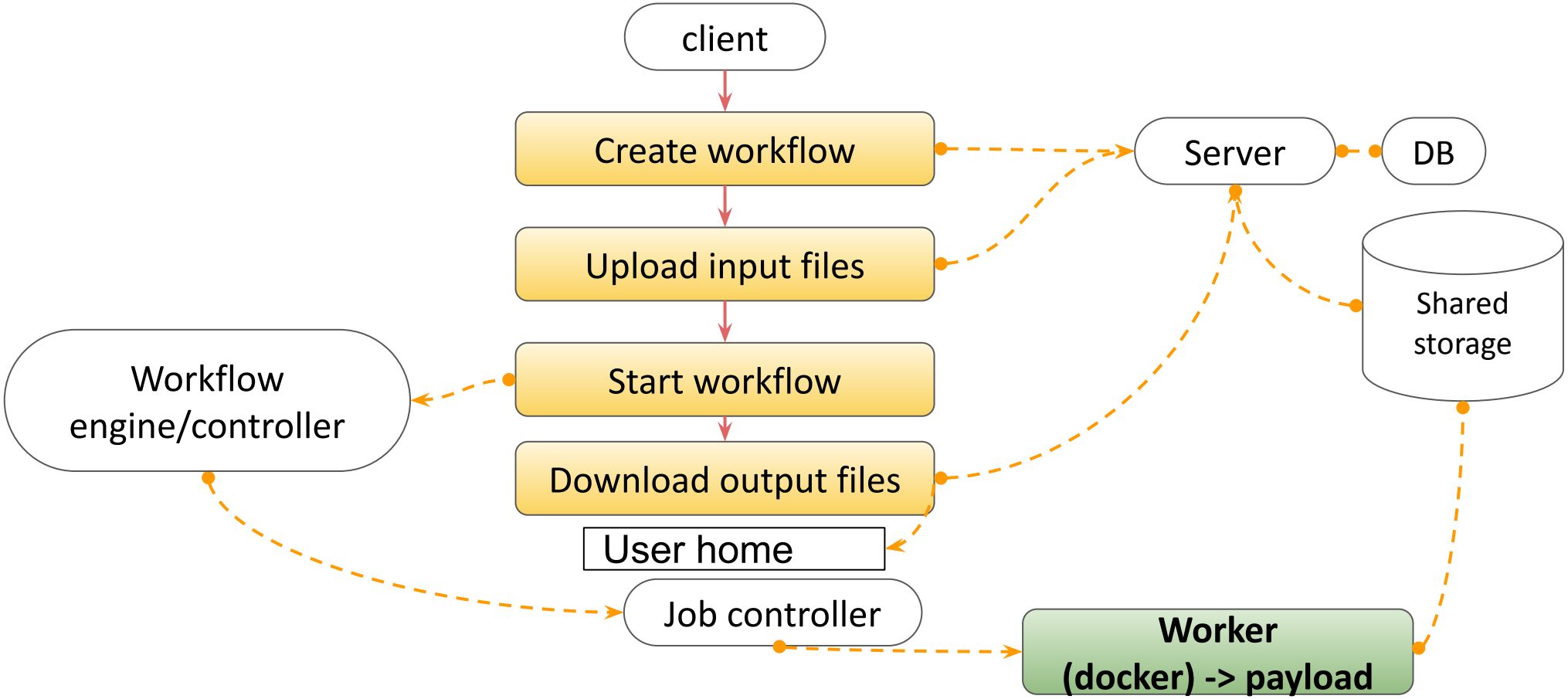
Regular private cluster

- Usually administered by the group/Institution
- Ability to install what you need (root access to worker nodes)
- Container orchestration technologies are easy to deploy
  - K8s, docker containers, etc.
- Long running processes on the submit node are usually allowed.
- ***Typically a few hundreds to a few thousand cores***

HPC centers

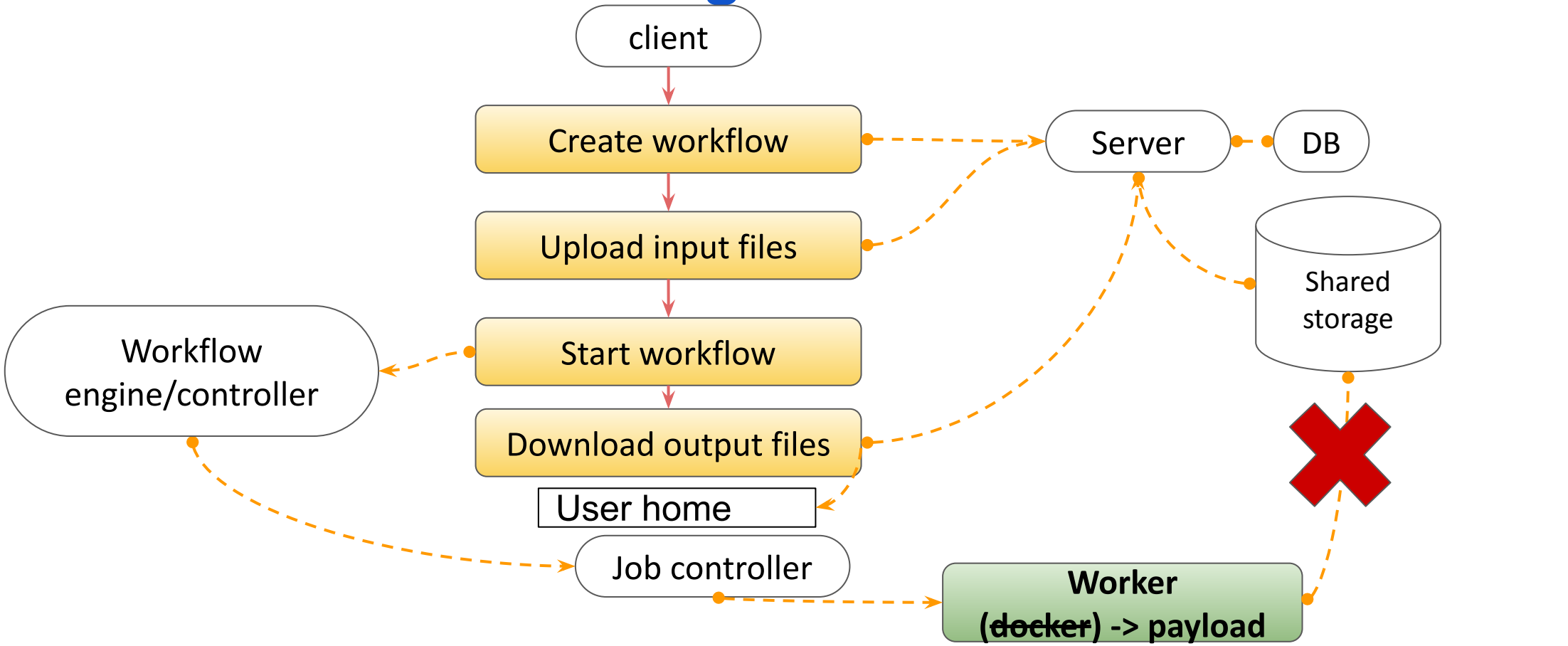
- No admin privileges to the resource
- Installing software dependencies that are not common can be challenging due to the lack of root privileges.
- Docker containers are NOT allowed, but they have their own options (e.g.: singularity, shifter), which run at the user level
- Long running processes on the submit node are usually not allowed.
- ***Hundreds of thousands of cores available***

REANA client / User's perspective
Starting a workflow



`/reana/users/<uuid>/workflows/<id>`

REANA client / User's perspective
Starting a workflow

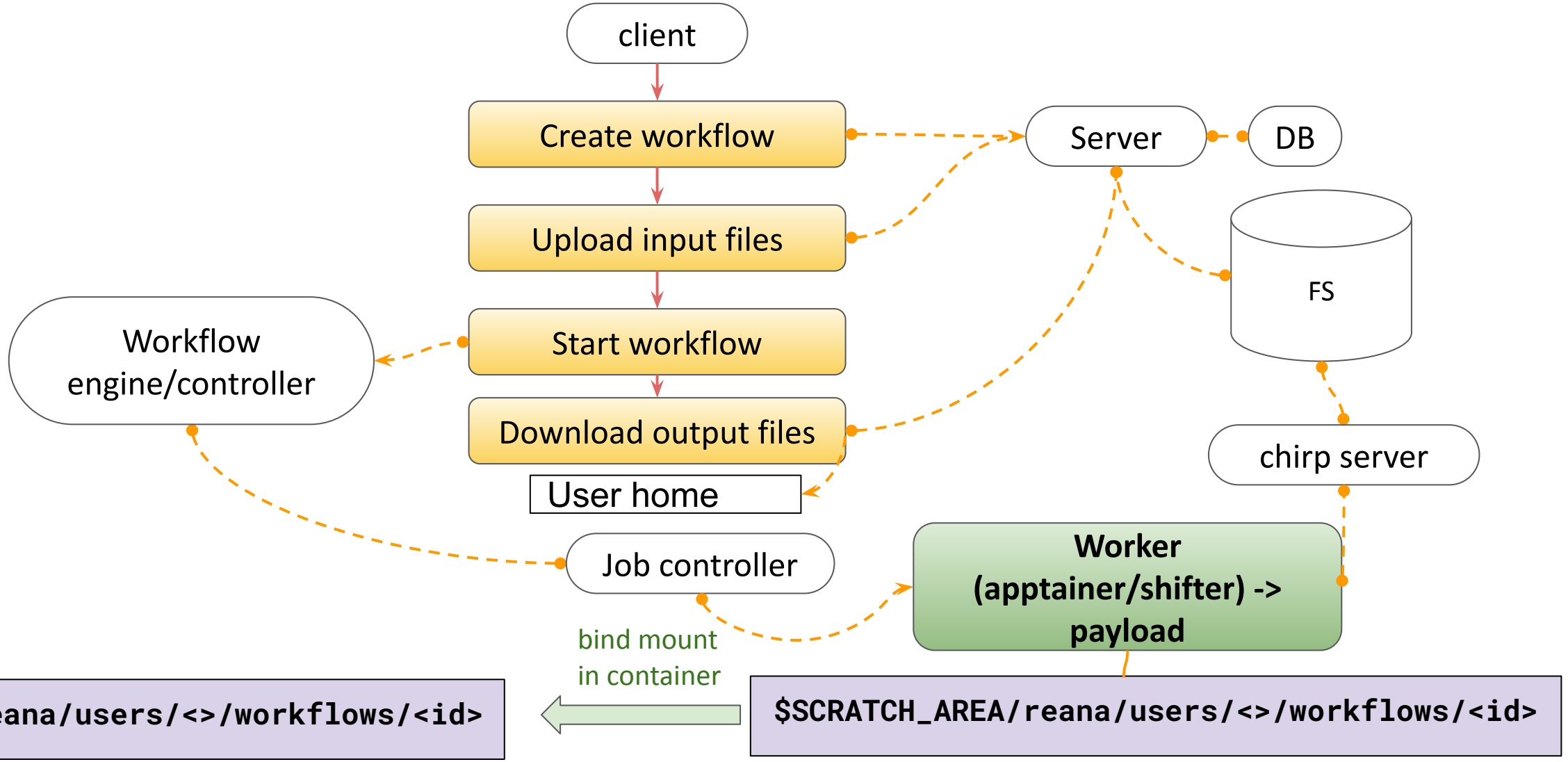


Notre Dame CCTools

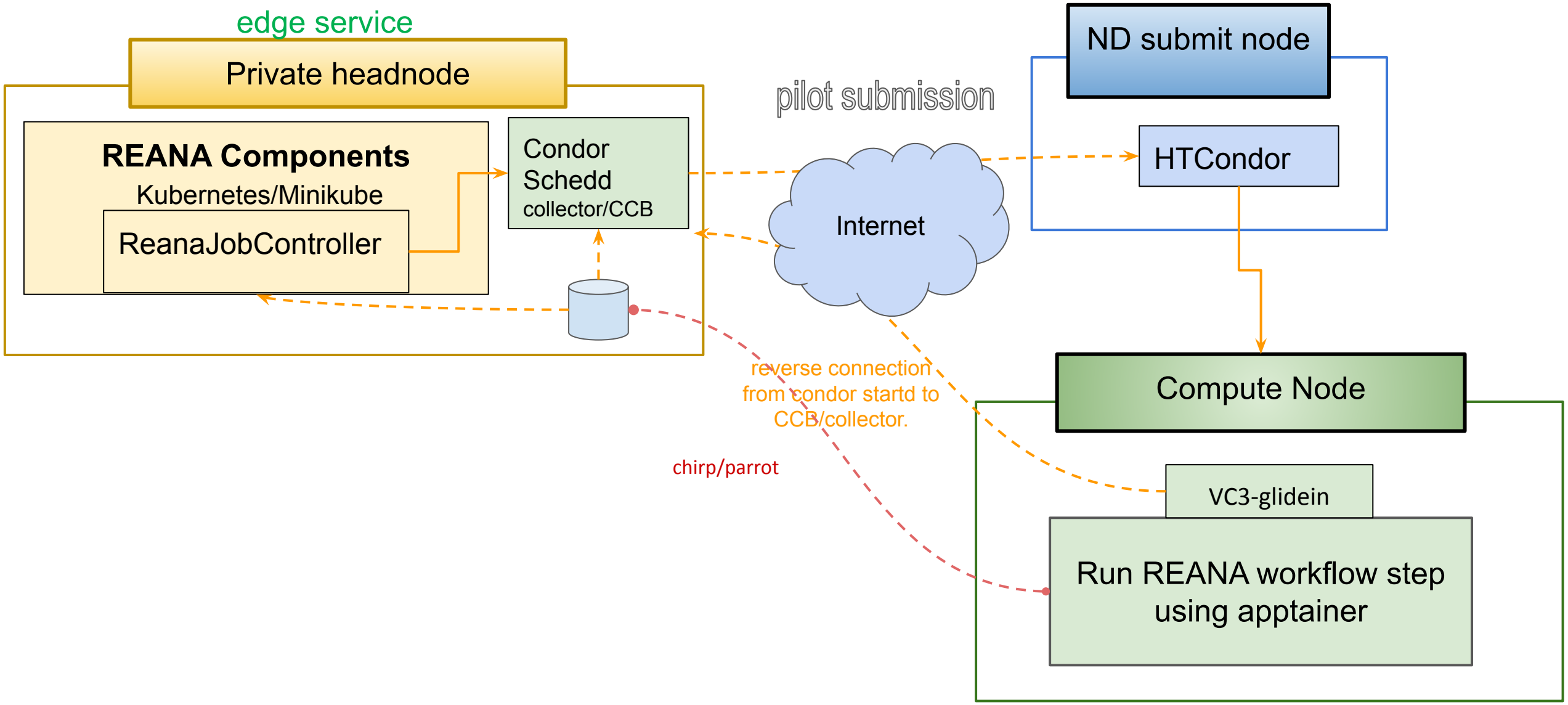
- **Chirp:** Lightweight user-level FS for collaboration across distributed systems such as clusters, clouds, and grids. An ordinary user can share storage space and data without requiring any sort of administrator privileges anywhere. Supports multiple authentication mechanisms.
- **Parrot:** A tool for attaching existing programs to remote I/O systems through the filesystem interface. E.g.:

  - `$ parrot_run vi /chirp/server.nd.edu/mydata`
  - `$ parrot_run cp /path/file /chirp/server.nd.edu/file`

REANA client / User's perspective
Starting a workflow



SCAILFIN @ND


edge service
Private headnode
REANA Components
Kubernetes/Minikube
ReanaJobController
Condor
Schedd
collector/CCB
pilot submission
Internet
ND submit node
HTCondor
reverse connection
from condor startd to
CCB/collector.
chirp/parrot
Compute Node
VC3-glidein
Run REANA workflow step
using apptainer