# Analysis Grand Challenge at REANA

Andrii Povsten

*Mentors:* Alex Held, Matthew Feickert (University of Wisconsin- Madison), Oksana Shadura (University Nebraska-Lincoln), Tibor Simko (CERN)

FAIR and Open Science in High Energy Physics, OAC-2226378, OAC-2226379 and OAC-2226380

# The Analysis Grand Challenge (AGC) project

The "Analysis Grand Challenge" (AGC) aims to help address the computing challenges of the HL-LHC

- coordinated by IRIS-HEP: research and development for HL-LHC (https://iris-hep.org/)
- organized jointly with the US ATLAS & US CMS operations programs

The AGC has **two aspects**:

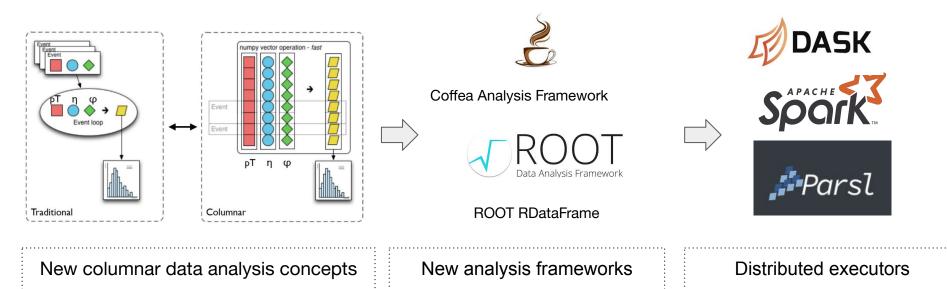
- 1. define a physics analysis task of realistic scope & scale
- 2. develop analysis pipelines that implements the task
  - find & address performance bottlenecks & usability concerns







# Supporting new data analysis concepts for HL-LHC



# The AGC physics analysis task

Main AGC analysis task: ttbar cross-section measurement

o using CMS Open Data (reformatted to 2 TB of NanoAODs): anyone can participate

o key feature: different kinds of systematic uncertainties & metadata handling

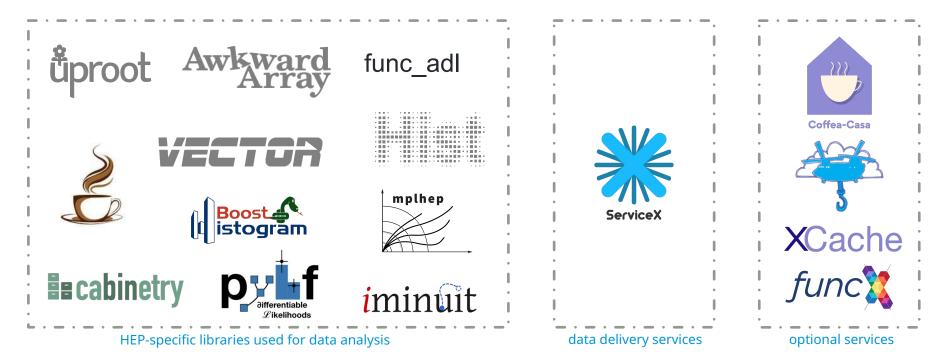
o sufficient complexity to demonstrate distributed scale-out performance

-	Auffest of units age and a set laterative heitigenets .		· A Patient Conternal
	ADDRESS THE PER ACTOR		
	BARDWOOD TUROS	# an # antigen # anne ant product all, the histogramic/liseable	
	distant and and and	<ul> <li>reters must white probability throughout this to a substantial through the second stant</li> </ul>	
	CLUTCH MAP	# all_histoprom + augentia.tertprodum_pl1_file_histoprom(10/per/))	
	ALLEND MURITY	wild FRMLIME "services, detailable":	
	COMPLET THE MERICE	A needs a signify extravel advect an according configuration of an extracting and according to the second s	
	640	print/Property had from the line and li	
	2000071000		
-	AN ADDRESS OF	Mandanado Isaa - Alfa	4/4 (00.14)
-	and a second second	Data devide loss. Resentantes 12%	1/10 (20134)
-	and respects	Page proved true, etc.	4/14 (141-24)
-	and the second s	The institute in the last and the	A/14 (14-15)
	NUMBER OF STREET	presidential land and a second se	And one-last
-	are stored		
	and the second se	Jings Journal Intel. Standard Mith.	4/10 (per 64)
	MARKS.	Man Jonate State, Wh	101-00-00
		Plan (month tion, inscriminal the second sec	141.000-000
	Manager 1	Physical Articles (Charles and Charles and	200.000.000
	and he are the test of the	Impolated-transformed 200	140.00.00
-	1404 1140.00		
	and and a second s	Jinge, Universite Annual State	275 (10-10)
-	ACCOUNTS IN COLUMNS	(Max Jonate Line , Transmitted 20%	201-00-00
-	ACCOUNT ON A DECEMBER OF	Papeland in the second se	4/10 (01/17)
=	and an an and a second	Mig-bashing Statistics (1)	1010 (00.01)
-	and and an and a second second	Name and Arrows and Arrows	VM DR DR
=	Automite tel'acteur	Phys. Americana Ch.	Vis po las
	Annual Science Texasters 1		
-		, Marchentities, 20	Aut Decisi
-	see 17 - NEW	Dispersive law intervention (1)	1/4 (00-24)
101. N	HTCHNOR Cluster	Inspecting the produced histograms	
-	and a support of the Designation of Designation of the second state of the second stat	and the free sector of the sec	



## Tools and services in IRIS-HEP AGC implementation

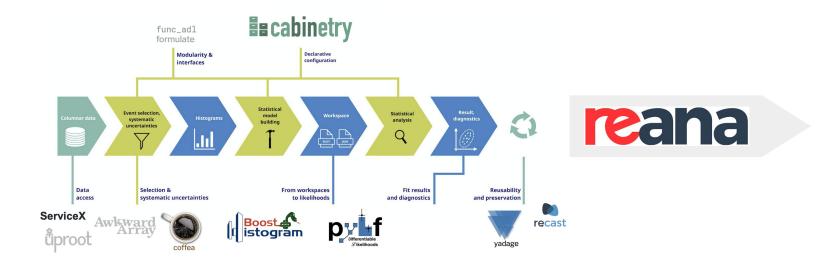
- Employing stack of Python HEP libraries for analysis tasks
- ServiceX used as data delivery service
- Execution on a coffea-casa analysis facility and NOW @ REANA



## Analysis Grand Challenge IRIS-HEP implementation

- Columnar data extraction from large dataset
- Processing of that data (event filtering, construction of observables, evaluation of systematic uncertainties) into histograms
- Statistical model construction and statistical inference
- Relevant visualisation for this steps

## + Adding analysis preservation step to AGC pipeline

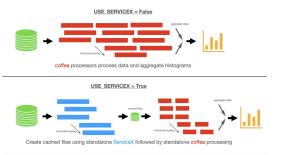


## Implementation: ttbar analysis in a notebook

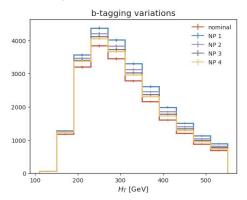
reconstructed observables

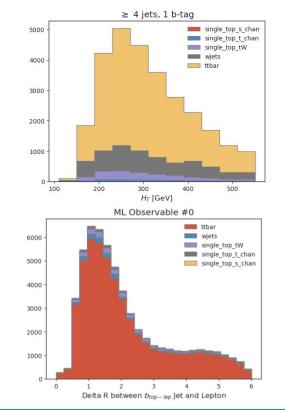
### From data delivery to statistical inference in a notebook

#### multiple supported processing schemes

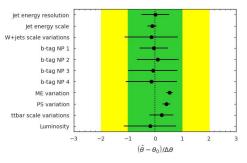


#### systematic variations

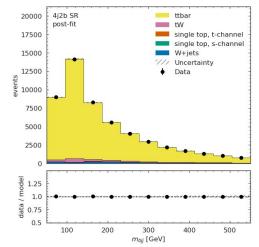




#### nuisance parameter pulls



#### post-fit distributions

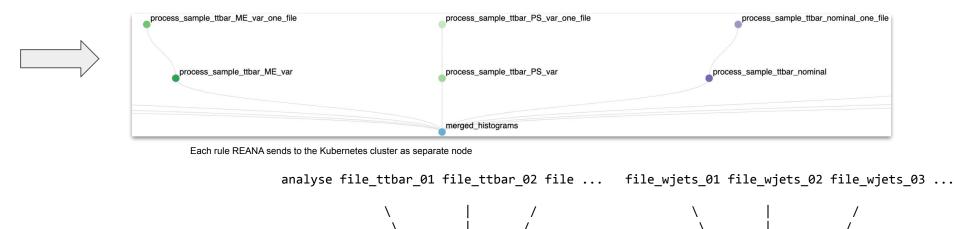




## Porting Coffea analysis to Snakemake

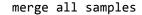
- Our choice was to use **Snakemake workflow management system (integrated in REANA)**
- Snakemake key feature is a "rule" description, which enables the parallelisation within REANA, running each rule in a separate pod.
- <u>Snakemake allows you to create a set of rules, each one defining a "step" of your</u> <u>analysis.</u>
- In AGC case we defined each step as processing one of AGC sample (9 in total) with output file containing processed histograms for given sample
- We end up having 2 rules for one sample and final merging rule, so in total we have 19 rules which would generate 788 jobs.

## Analysis Grand Challenge pipeline: Adapting to Snakemake







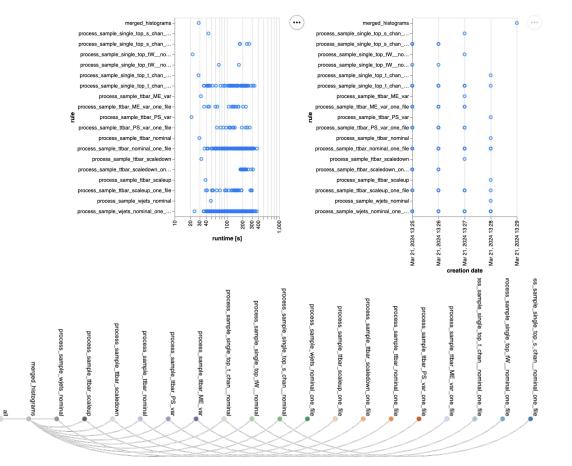


Snakemake checks the inputs and outputs in the rules to see the dependencies and order of execution

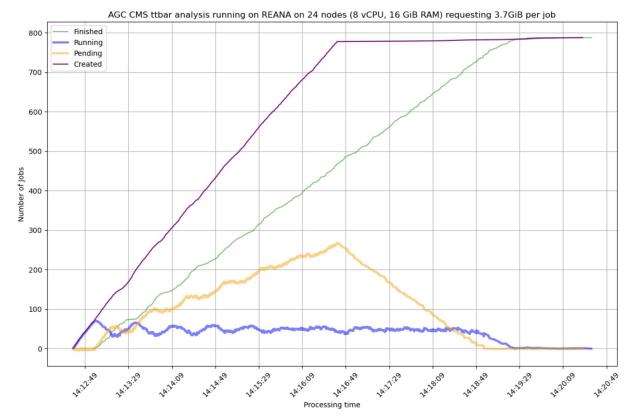
## Examples of Snakemake rules

```
rule process_sample_ttbar_nominal:
                                                                                              rule process_sample_ttbar_nominal_one_file:
                                                                                                container:
  container:
    "ttbarkerberos:20240311""
                                                                                                   "ttbarkerberos:20240311"
  resources:
                                                                                                resources:
    kubernetes_memory_limit="3700Mi"
                                                                                                   kubernetes_memory_limit="3700Mi",
                                                                                                   kerberos = True
  input:
    "file_merging.ipynb",
                                                                                                output:
                                                                                                   "histograms/histograms_ttbar__nominal_{filename}"
    expand(get file paths("ttbar nominal"))
  output:
                                                                                                params:
    "everything merged ttbar nominal.root"
                                                                                                  sample name = 'ttbar nominal'
                                                                                                shell:
  params:
    sample_name = 'ttbar__nominal'
                                                                                                   "/bin/bash -I && source fix-env.sh && python prepare_workspace.py
                                                                                              sample {params.sample name} {wildcards.filename} && papermill
  shell:
    "papermill file merging.ipynb merged nominal.ipynb -p sample name
                                                                                              ttbar analysis reana.ipynb sample {params.sample name} {wildcards.filename} out.ipynb
{params.sample_name} -k python3"
                                                                                              -p sample_name {params.sample_name} -p filename {url_prefix}{wildcards.filename} -k
                                                                                              python3"
```

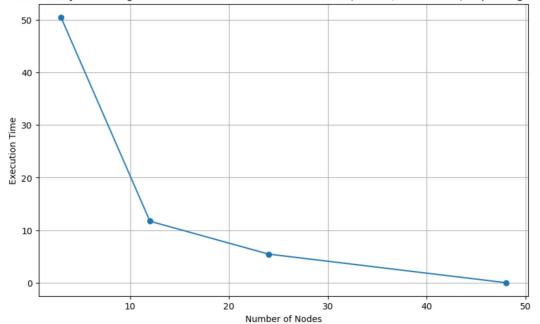
## **REANA AGC report**



# Optimising REANA k8s nodes configurations (depending on configuration)



## The current execution time statistics



AGC CMS ttbar analysis running on REANA on different number of nodes (8 vCPU, 16 GiB RAM) requesting 3.7GiB per job

# Issues with porting AGC IRIS-HEP implementation to be executed on REANA

- Defining efficient resources = optimisation process
- Specify the kubernetes\_memory limit that exact number of jobs are running in one node.
- XCache switched off
- Dask switched off
  - Datasets at Nebraska are too far for CERN REANA instance and accessing datasets from EOS Public is sometimes not efficient
- No native Dask support in REANA (WIP!)

## Conclusion

- We successfully implement the AGC ttbar analysis at REANA using Snakemake
- You need to develop the Snakemake skills to be able to make your analysis reproducible friendly.

## Future Tasks

- Add the recasting step using RECAST which would allow to submit, evaluate, of additional sample which could be then merged on the final step
- Making more stress test experiments of AGC based on job memory
- Testing AGC ServiceX and Machine Learning pipelines in REANA
- Make a clear instruction for the <u>reana-demo-agc-cms-ttbar-coffea</u>.



## AGC notebooks modification:

- Rerun the same notebook n-times but with different parameters => instead of processing all files, samples we process one sample with one file
- Firstly we parallelized each sample from fileset:

```
original_dict = fileset
selected_file = original_dict[sample_name]['files']
new_dict = {sample_name: {'files': [filename], 'metadata': original_dict[sample_name]['metadata']}}
```

- Second, parallelize each file for each sample:

```
all_histograms, metrics = run(
    fileset={sample_name: new_dict[sample_name]},
    treename=treename,
    processor_instance=TtbarAnalysis(USE_INFERENCE, USE_TRITON)
)
```

The main idea is to see the whole picture of your analysis what steps suppose to be after another and modify it on the early stages to have a separate pieces which could be count as 1 job.