Support for HTCondor high-throughput computing workflows in the **REANA** reusable analysis platform

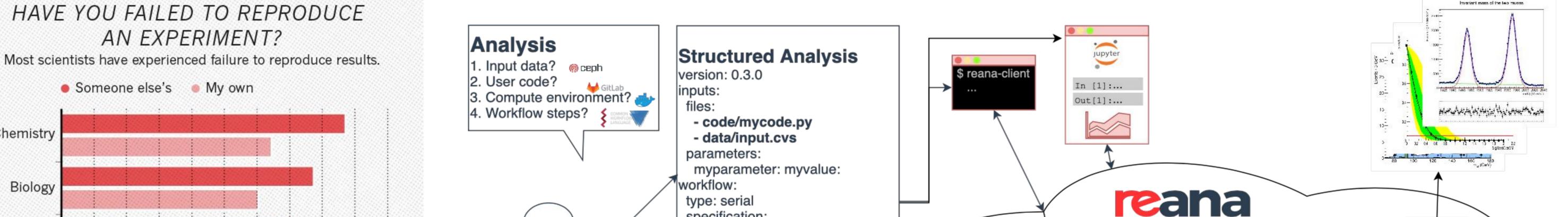
Rokas Mačiulaitis	Paul Brenner	Scott Hampton	Michael D. Hildreth	Kenyi Paolo Hurtado Anampa	Irena Johnson	Cody Kankel	Jan Okraska	Diego Rodriguez	Tibor Šimko
CERN	University of Notre Dame	University of Notre Dame	University of Notre Dame	University of Notre Dame	University of Notre Dame	University of Notre Dame	CERN	CERN	CERN
Switzerland	United States	United States	United States	United States	United States	United States	Switzerland	Switzerland	Switzerland
rokas.maciulaitis@cern.ch	paul.r.brenner@nd.edu	shampton@nd.edu	mhildret@nd.edu	khurtado@nd.edu	ijohnso1@nd.edu	ckankel@nd.edu	jan.okraska@cern.ch	diego.rodriguez@cern.ch	tibor.simko@cern.ch

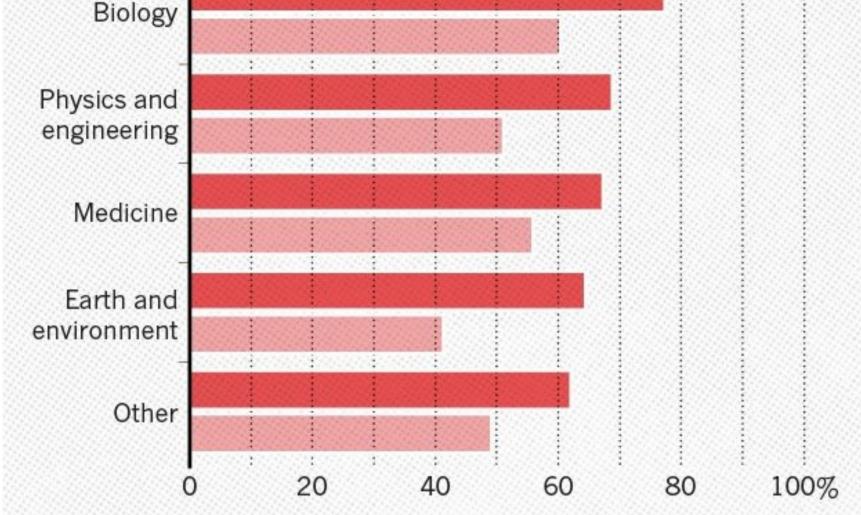
Reproducible research

REANA approach

REANA

Server

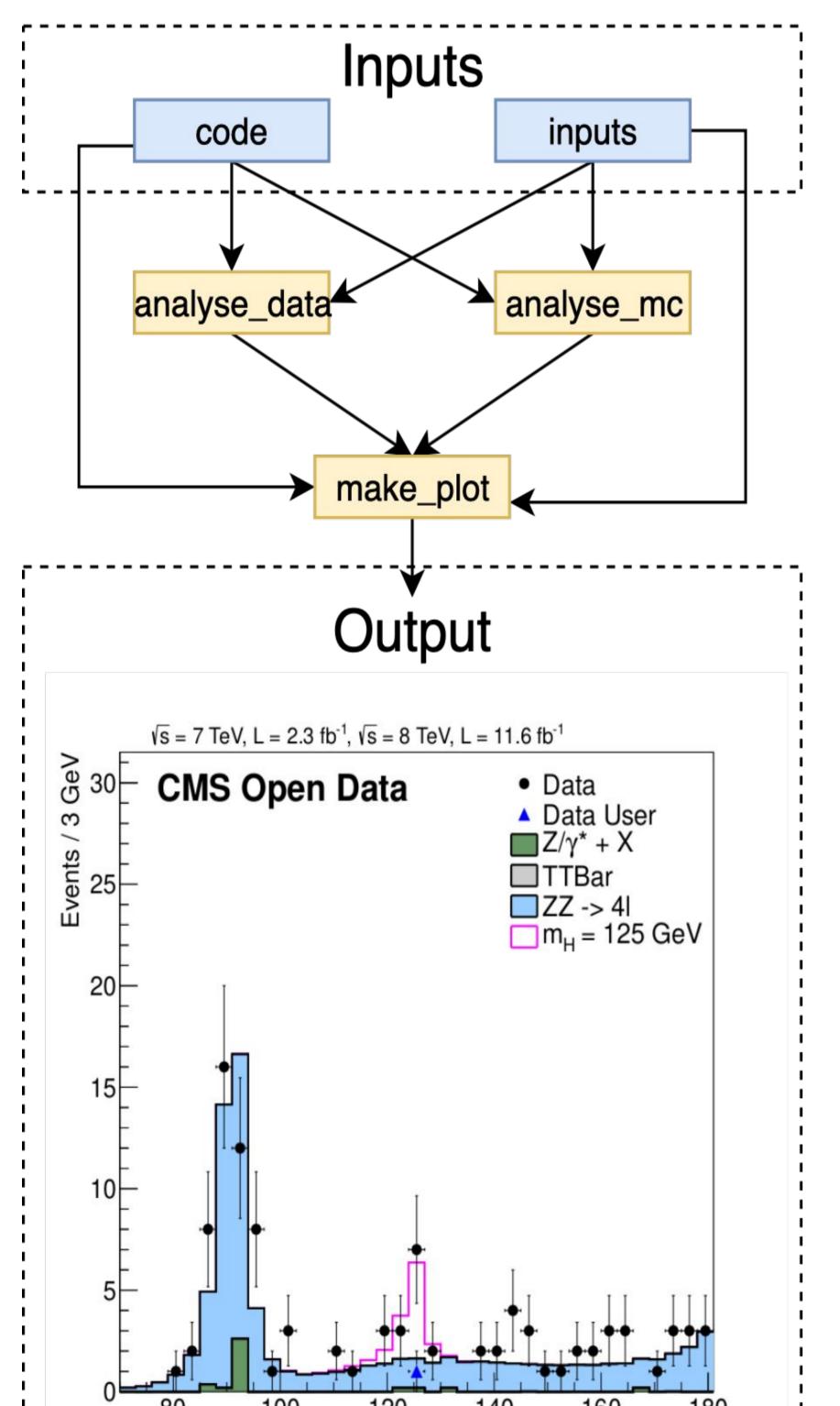


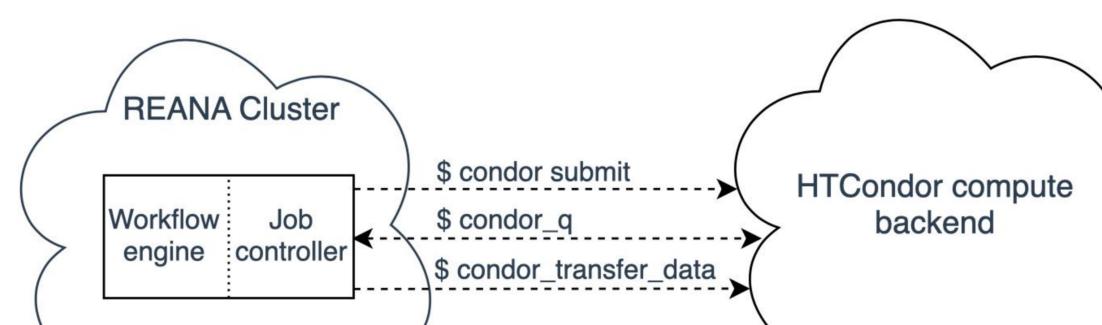


Chemistry

https://www.nature.com/news/1-500-scientists-lift-the-lid-onreproducibility-1.19970

Example workflow





type: serial

steps:

outputs:

files:

specification:

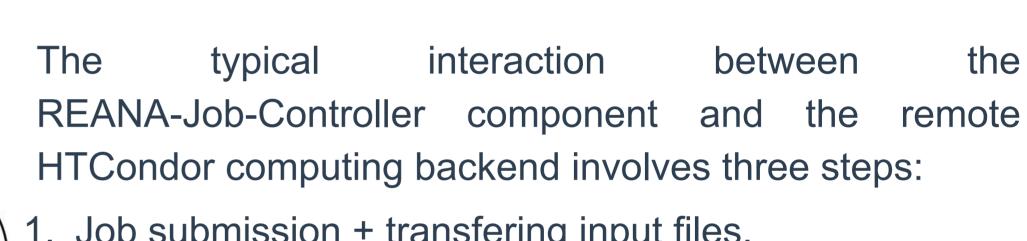
commands:

- results/greetings.txt

- environment: 'centos7'

- python code/mycode.py

--par "\${myparameter}"



#1

Job

#1

Job

#2

Job

#3

#2

Job

Job submission + transfering input files.

#1 Workflow

Engine

#2 Workflow

Engine

 \mathfrak{B}

kubernetes

→

Reana

Workflow

Controller

#1 Job

Controller

#2 Job

Controller

Job monitoring



Kubernetes

Scientist

HTCondor

Shared

storage

Extending REANA with HTCondor

Abstracting job submission

REANA platform submit The analysis workflow steps through the Job-Controller component which take container image, inputs, a commands to run to perform t workflow tasks. The step controller component then launch the job using the Kubernetes Job API We taken have existi **REANA-Job-Controller REST API** ar the design to extended suppo arbitrary compute backend such as HTCondor for high-throughput

submits	JobManager	The design
which takes age, and	+ docker_image: str + cmd: str + env_vars: dict + job_id: str + workflow_uuid: str + job_name: str	platform When a us is to b REANA-Jo
en launches es Job API. existing	+ execute() + stop() + get_status() + get_logs() + execution_hook() + _create_job_in_db() + _cache_job() + _update_job_status()	parameter backend. T means of analyses. operations further stud
KubernetesJobManager	HTCondorJobManager	SlurmJobManager
+ docker_image: str + cmd: str + env_vars: dict + job_id: str + workflow_uuid: str + job_name: str + compute_backend: str + cvmfs_mounts: str	+ docker_image: str + cmd: str + env_vars: dict + job_id: str + workflow_uuid: str + job_name: str + compute_backend: str	+ docker_image: str + cmd: str + env_vars: dict + job_id: str + workflow_uuid: str + job_name: str + compute_backend: str
+ shared_file_syste: bool + execute() + stop() + _add_shared_volume()	+ execute() + stop() + authenticate() + spool_output() + get_logs() + find_job_in_history() + _submit() + _get_schedd()	+ execute() + stop()

Validation

The designed solution was prototyped in the REANA platform using the CERN HTCondor cluster. When a user specifies that a certain workflow job run on the HTCondor backend, the be to IS REANA-Job-Controller container takes care of job parameter translation for the targeted compute backend. The developed prototype was tested by means of running several particle physics model analyses. The configurable level of "map-reduce" operations in the DAG workflow graph allows to further study the scalability of the solution.

We have furthermore integrated REANA with the Virtual Clusters for Community Computation (VC3) environment. We have developed Ansible templates allowing individual users deploy the to personal REANA system in a VC3 environment. The deployment from integration with a benefits high-throughput of range computing backend that the VC3 environment natively supports. Our thus validated by design was two independent of means deployment scenarios.



This REANA reproducible analysis example studies the Higgs-to-four-lepton decay channel that led to the Higgs boson experimental discovery in 2012. The example uses CMS open data taken in 2011 and 2012.

https://github.com/	reanahub/reana-demo-cms-h4l
<u> </u>	

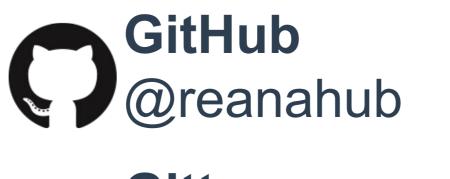
computing.	The
abstraction	regards
job submis	sion and
execution, j	ob status
monitoring,	and the
input/output	data
transfer	amongst
supported ba	ackends.

computing or Slurm for

high-performance

+ job_id: str + workflow_uuid: str	
+ job_name: str	
+ compute_backend:	str
+ execute()	
+ stop()	

www.reana.io



Gitter gitter.im/reanahub/reana

