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

Rokas Mačiulaitis  
CERN  
Geneva, Switzerland  
rokas.maciulaitis@cern.ch

Paul Brenner  
University of Notre Dame  
Notre Dame, Indiana, USA  
paul.r.brenner@nd.edu

Scott Hampton  
University of Notre Dame  
Notre Dame, Indiana, USA  
shampton@nd.edu

Michael D. Hildreth  
University of Notre Dame  
Notre Dame, Indiana, USA  
mhildret@nd.edu

Kenyi Paolo Hurtado Anampa  
University of Notre Dame  
Notre Dame, Indiana, USA  
khurtado@nd.edu

Irena Johnson  
University of Notre Dame  
Notre Dame, Indiana, USA  
lijohnson@nd.edu

Cody Kankel  
University of Notre Dame  
Notre Dame, Indiana, USA  
ckankel@nd.edu

Jan Okraska  
CERN  
Geneva, Switzerland  
jan.okraska@cern.ch

Diego Rodríguez  
CERN  
Geneva, Switzerland  
diego.rodriguez@cern.ch

Abstract—REANA is a reusable and reproducible data analysis platform allowing researchers to structure their analysis pipelines and run them on remote containerised compute clouds. REANA supports several different workflows systems (CWL, Serial, Yadage) and uses Kubernetes’ job execution backend. We have designed an abstract job execution component that extends the REANA platform job execution capabilities to support multiple compute backends. We have tested the abstract job execution component with HTCondor and verified the scalability of the designed solution. The results show that the REANA platform would be able to support hybrid scientific workflows where different parts of the analysis pipelines can be executed on multiple computing backends.

Index Terms—reproducible science, computational workflows, high-throughput computing, high-performance computing, data analysis

I. INTRODUCTION

REANA [1], [2] is a reproducible and reusable data analysis platform allowing researchers to specify computational workflow steps in several declarative languages (such as CWL [3] or Yadage [4]) to run their data analysis pipelines on containerised compute clouds. The platform was developed with the data analysis reproducibility and reusability in mind [5]. The overview of the REANA platform is presented in Fig. 1.

The computational workflows associated to research data analyses can be represented in the form of a Directed Acyclic Graphs (DAG) that topologically order the analysis steps and define their dependencies. The DAG workflows may consist of running several tens of thousands of computational tasks. The REANA platform was developed with the Kubernetes-orchestrated job execution backend as the primary execution platform. In order to extend high performance and scalability of the platform, the present work introduces a ability to run workflow tasks on the HTCondor backend [6].

II. JOB CONTROLLER ABSTRACTION

The REANA platform submits analysis workflow steps through the REANA-Job-Controller component which takes inputs, container image, and commands to run to perform the workflow step tasks. The job controller component then launches the job using the Kubernetes Job API.

We have taken existing REANA-Job-Controller REST API and extended the design to support arbitrary compute backend such as HTCondor for high-throughput computing [6] or Slurm for high-performance computing [7]. The abstraction regards job submission and execution, job status monitoring, and the input/output data transfer amongst supported backends and is presented in Fig. 2.
The abstract REANA-Job-Controller component interface has been designed to allow the job execution, the job monitoring, and the input/output file transfer with support for multiple compute backends. The execution\_hook() decorator function allows for easy customisation of the interface with ad hoc “pre-” and “post-” submission tasks suitable for each concrete backend.

### III. Prototype

The designed solution was prototyped in the REANA platform using the CERN HTCondor cluster deployment. When a user specifies that a certain workflow job is to be run on the HTCondor backend, the REANA-Job-Controller container takes care of job translation and communication with the target compute backend. The typical communication scenario between the Kubernetes cluster and the HTCondor cluster is presented in Fig. 3.

The developed prototype was tested by means of running several particle physics model analyses [8]. 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 [9]. We have developed Ansible templates allowing individual users to deploy the personal REANA system in a VC3 environment. The deployment benefits from integration with a range of high-throughput and high-performance computing backends that the VC3 environment natively supports. Our design was thus validated by means of two independent deployment scenarios.

### IV. Conclusions

We have extended the REANA architecture to allow the execution of analysis jobs on arbitrary compute backends. We have prototyped the solution for running high-throughput scientific workflows using the HTCondor backend. We have implemented support for CERN HTCondor deployment and VC3 deployment scenarios. The developed solution was validated by means of several particle physics model analyses. The present work lays foundation towards a possibility of running hybrid scientific workflows where different part of the analysis pipeline can be executed on different computing backend.

### ACKNOWLEDGMENT

This work was partially supported by the United States’ National Science Foundation grant “Scalable CyberInfrastructure for Artificial Intelligence and Likelihood Free Inference” (SCAILFIN).

### REFERENCES

7. Slurm workload manager. https://slurm.schedmd.com/