Scalable ATLAS pMSSM computational workflows using containerised REANA reusable analysis platform

Marco Donadoni[1], *Matthew* Feickert[2], *Lukas* Heinrich[3], *Yang* Liu[4], *Audrius* Mečionis[1], *Vladyslav* Moisieienkov[1], *Tibor* Šimko[1,*], *Giordon* Stark[5], and *Marco* Vidal García[1]

[1]CERN, Geneva, Switzerland
[2]University of Wisconsin-Madison, Madison, United States
[3]Max-Planck-Institut für Physik, München, Germany
[4]Sun Yat-Sen University, China
[5]SCIPP, UC Santa Cruz, United States

Abstract. In this paper we describe the development of a streamlined framework for large-scale ATLAS pMSSM reinterpretations of LHC Run-2 analyses using containerised computational workflows. The project is looking to assess the global coverage of BSM physics and requires running O(5k) computational workflows representing pMSSM model points. Following ATLAS Analysis Preservation policies, many analyses have been preserved as containerised Yadage workflows, and after validation were added to a curated selection for the pMSSM study. To run the workflows at scale, we utilised the REANA reusable analysis platform. We describe how the REANA platform was enhanced to ensure the best concurrent throughput by internal service scheduling changes. We discuss the scalability of the approach on Kubernetes clusters from 500 to 5000 cores. Finally, we demonstrate a possibility of using additional ad-hoc public cloud infrastructure resources by running the same workflows on the Google Cloud Platform.

1 Introduction

We have developed a streamlined framework for large-scale pMSSM reinterpretations of ATLAS analyses of LHC Run-2 using containerised computational workflows. The project is looking to assess the global coverage of BSM physics and requires running numerous computational workflows representing pMSSM model points. The framework builds upon the idea of RECAST-ing analyses [1] and takes into account the experiences with the previous ATLAS pMSSM reinterpretations from LHC Run-1 period [2].

Following the ATLAS analysis preservation policies, many ATLAS analyses have been preserved as containerised Yadage workflows. After validation they are added to a curated selection of analyses suitable for the pMSSM study. Figure 1 shows one such repository for the supersymmetry searches.
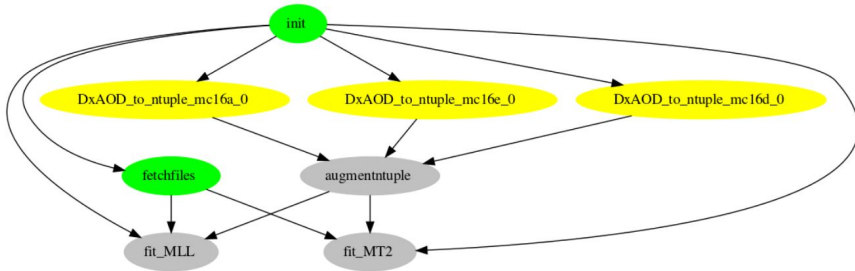
One typical pMSSM computational workflow is presented in Figure 2. The workflow consists of three time-consuming ntupling steps that process data files and run in parallel. The workflow ends with a latter fitting steps that run afterwards. The dependency of steps in
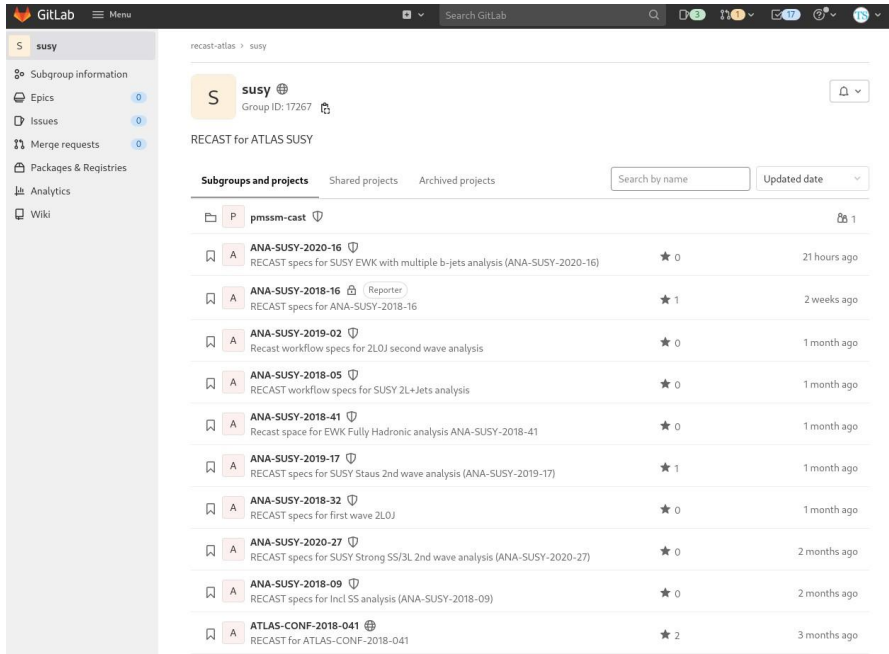

Figure 1. A screenshot of the ATLAS SUSY group analyses preserved on GitLab. Each repository is labeled with the internal ATLAS analysis identifier and contains both workflow files and additional data files needed for the computational processing.

Figure 2. A typical pMSSM workflow. The computational runtime is about 10 minutes without systematics (test payload) and about 10 hours with all systematics (real payload).

the computational graph is rather simple. The complexity of the problem lies in having to run several thousands of these workflows in order to cover a sufficient number of pMSSM model points.

It was the goal of the present work to study the feasibility of running several thousands of these containerised workflows in parallel in an automated way in order to facilitate typical pMSSM studies.
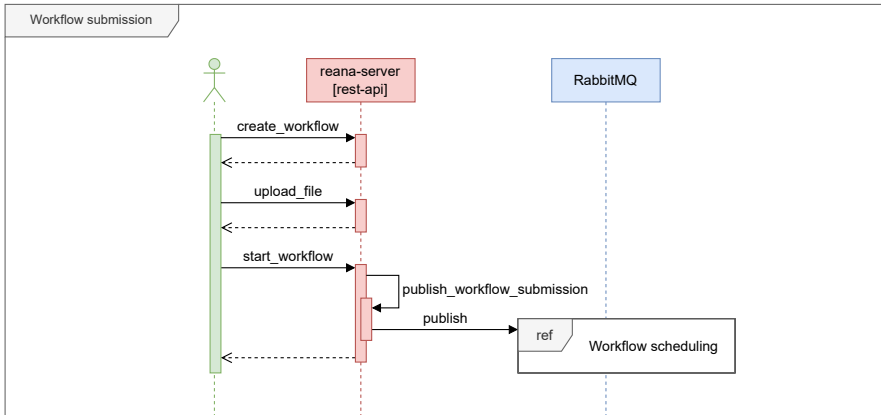
Figure 3. The sequence diagram showing how REANA schedules incoming workflows after submission. The submitted workflows are announced via message queue that is later processed by the workflow scheduler in Figure 4.

2 Method

The computational workflows were run at scale using the REANA reusable analysis platform [3]. The computational backend was the Kubernetes cluster of various sizes (from 500 cores up to 5000 cores). We have been varying several parameters of the cluster such as the number of nodes and the required memory and studied the maximum number of pMSSM workflows that the platform can handle concurrently. After performing several such computational experiments, we have improved the scheduling efficiency of REANA to increase the running bandwidth for the pMSSM style of workflows.

Figure 3 shows the sequence diagram of the workflow submission stage. The incoming workflows are stored in a queue that is later processed by the scheduler. The first task was to improve the performance of the REANA platform's server submission end points to allow many concurrent workflow starting requests.

Figure 4 shows the next stage of the process, namely how the submitted workflows are being consumed from the incoming queue. The scheduler first checks whether the incoming workflow does not exceed the limits on the total number of workflow the system could handle as well as currently available free memory on the Kubernetes cluster. If the checks succeed, the workflow is accepted for execution. In the opposite case the incoming workflow is being rescheduled and attempted to be accepted for execution several times whilst waiting for the Kubernetes cluster resources to liberate. If the workflow cannot be scheduled for a substantial amount of time, a failure is declared.

Figure 5 shows the stage of the running of the workflow after it has been accepted for execution. Note the interplay of the REANA platform with the underlying Kubernetes cluster: the job is scheduled using the Kubernetes native job scheduler mechanism which include additional scheduling delays that needed to be taken into account for optimisation. The progress of the workflow is monitored until the workflow execution terminates. The workflow steps are launched when the worker nodes are free to run the workload. The status of jobs is published in the message queue.

Figure 6 shows the termination stage of the workflow. When all the steps are finished and the results are produced, the system has to delete the Kubernetes pod and update the status of the workflow in both the message queue and the database. This constituted another layer
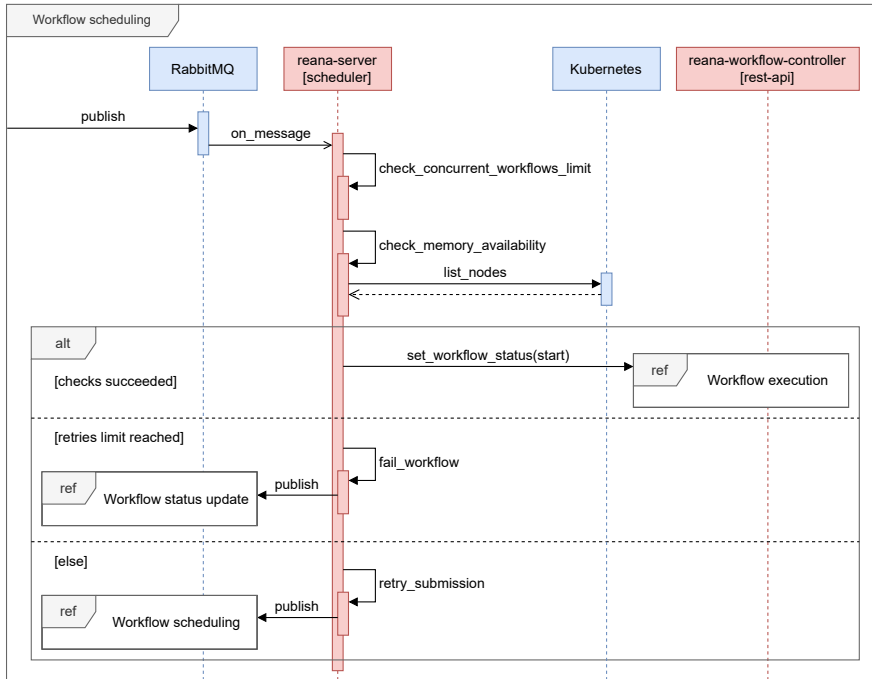
Figure 4. The sequence diagram showing how REANA schedules queued workflows. The system checks for available resources before allowing workflow runs for execution. The checking and rescheduling workflow offers several possibilities for optimisations. The workflows accepted for execution are further processed in Figure 5.

of optimisations in order to handle any status handling processes in an asynchronous manner whilst the platform is starting the new incoming workflows.

3 Results

We have improved the REANA platform scheduling performance in order to maximise the scheduling throughput of incoming workflows at the various stages of the workflow life cycle as described in Section 2. A special attention was paid to measure the CPU and Memory usage of the cluster nodes.

Figure 7 shows a typical snapshot of the status of cluster nodes running the pMSSM workloads. We have used nodes of the `m2.xlarge` flavour which consist of 16 GiB of available memory and 8 virtual cores. One can see the efficient use of cores of the cluster resulting from tuning REANA parameters such as the number of nodes running workflow orchestration tasks, the number of nodes running the pMSSM workflow step jobs themselves, as well as the memory request limits for each ntupling job of the first pMSSM workflow stages.

Figure 8 shows the results of one of our scalability experiment that consisted of submitting 200 new pMSSM workflows every 10 minutes. A cluster with 448 cores presented on the left cannot keep up with such a workload: note the increasing scheduling waiting times (plotted in the orange colour) as well as increasing workflow run times (plotted in blue). The overflow happens because the cluster is allowing more workflows than it can hold. However, note how
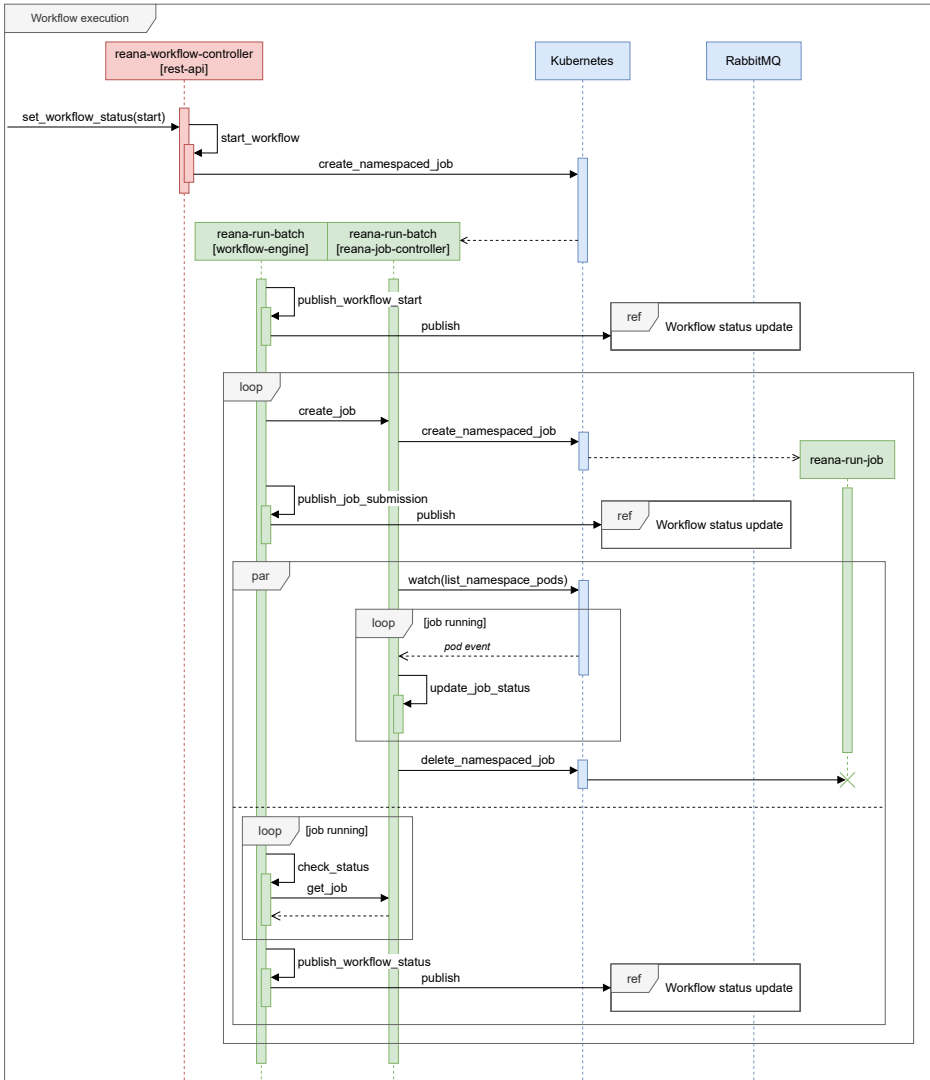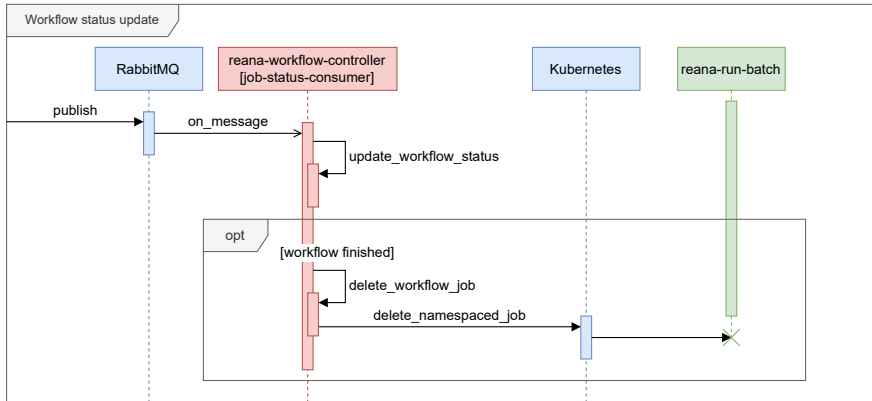
Figure 5. The sequence diagram showing how the REANA executes scheduled workflows. Note the interplay between the scheduler and the Kubernetes cluster. The pod creation offers another space for optimisations. The workflow execution status monitoring is carried out by a watching loop. The workflow jobs are started for each workflow step. The termination procedures are further illustrated in Figure 6.

the same cluster with 1072 cores presented on the right of the Figure holds the same workload very comfortably.

Figure 9 shows the same kind of experiment executed over a longer period of time. This helped to ensure that the platform can sustain the constantly increasing stream of incoming workloads.

We have run several benchmarking experiments in the CERN Computer Centre and, to test the portability, performed a few runs also on the Google Cloud Platform. This allowed

Figure 6. The sequence diagram showing how REANA updates workflow statuses and terminates finished workflows. The procedure involves consuming the message queue, closing the Kubernetes pods, and updating the database about the status of the workflow run. In case of launching several thousands of concurrent workflows, these processes also have to be optimised.

```
$ kubectl top nodes
NAME                                CPU(cores)    CPU%    MEMORY(bytes)    MEMORY%
reanaatlas1-3slyowp42qex-node-15    7858m         98%     12033Mi          82%
reanaatlas1-3slyowp42qex-node-16    7848m         98%     12083Mi          83%
reanaatlas1-3slyowp42qex-node-17    7846m         98%     12210Mi          83%
reanaatlas1-3slyowp42qex-node-18    7773m         97%     8995Mi           61%
reanaatlas1-3slyowp42qex-node-19    7864m         98%     11516Mi          79%
reanaatlas1-3slyowp42qex-node-20    7843m         98%     12177Mi          83%
reanaatlas1-3slyowp42qex-node-21    7376m         92%     8698Mi           59%
reanaatlas1-3slyowp42qex-node-22    7817m         97%     11201Mi          77%
reanaatlas1-3slyowp42qex-node-23    7748m         96%     9978Mi           68%
reanaatlas1-3slyowp42qex-node-24    7854m         98%     12161Mi          83%
reanaatlas1-3slyowp42qex-node-25    7868m         98%     12293Mi          84%
reanaatlas1-3slyowp42qex-node-26    7787m         97%     10991Mi          75%
```

Figure 7. An example of the benchmark tests running in the CERN Computer Centre. The REANA scheduling parameters were optimised to maximise the CPU utilisation and the Memory consumption on the cluster for the typical pMSSM ntupling job parallelism (see Figure 2). Note the very good efficiency of CPU cores in the above screenshot.
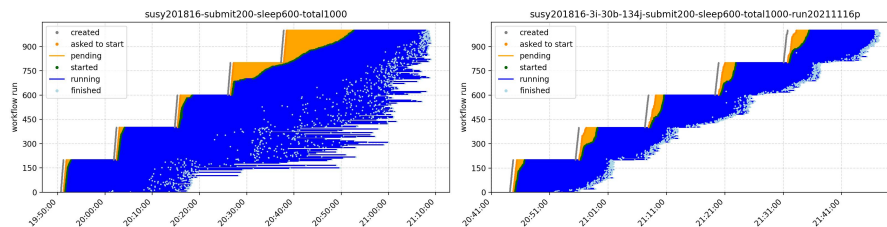
Figure 8. A scalability test submitting 200 workflows every 10 minutes. A cluster with 448 cores (left) cannot keep up with the load. A cluster with 1072 cores (right) can comfortably hold the incoming workload.
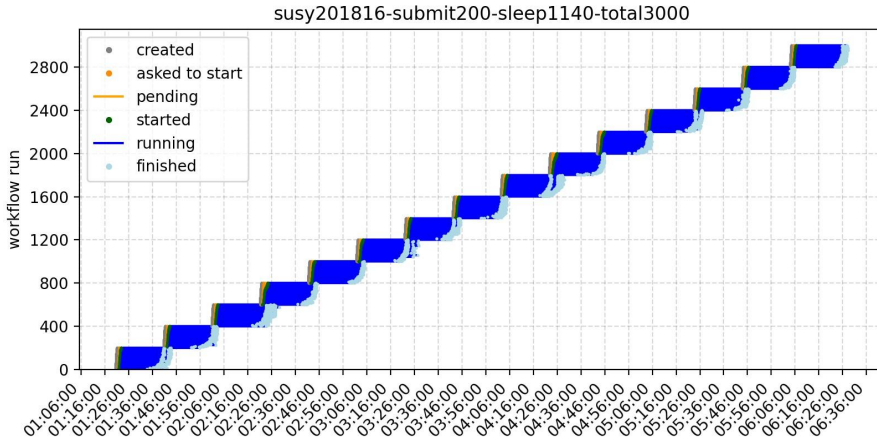
Figure 9. The workload burndown throughput rate is sustainable over a long period of time.

to prove the applicability of the approach on various compute backends, facilitating future reproducibility of containerised workflows irrespective of their original computing environments.

4 Conclusions

ATLAS searches for new physics are being effectively preserved together with containerised computational workflow recipes as part of the ATLAS RECAST project. This enables their future reuse and reinterpretation and greatly facilitates the running of efficient pMSSM studies over a large collection of individual analyses.

We have launched several ATLAS pMSSM workflows on the REANA reproducible analysis platform and studied the performance from workflow scheduling up to workflow execution and termination procedures with the aim of allowing running several thousands of these workflows to cover a sufficient number of pMSSM model points.

The REANA platform has been internally optimised to allow faster workflow scheduling, processing and terminating procedures on an individual workflow level as well as under the stressing conditions of processing many incoming concurrent workloads. A set of benchmarking experiments allowed to optimise and tune the REANA system for the pMSSM workloads on the Kuberentes clusters ranging from medium to large sizes (from 500 to 5000 cores). It was essential to adjust REANA scheduling parameters to the type of the pMSSM workload in order to ensure the best throughput and the efficient cluster CPU and memory resource utilisation.

The developed system was tested on the CERN Computer Centre as well as on the Google Cloud Platform in order to ensure the reproducibility of the approach and is fully ready to run large-scale ATLAS pMSSM reinterpretations of LHC Run-2 analyses. The first results by the ATLAS collaborations are being published [4].

Acknowledgements

References

[1] K. Cranmer, I. Yavin, "RECAST: Extending the Impact of Existing Analyses", JHEP 1104:028 (2011), `https://doi.org/10.48550/arXiv.1010.2506`.

[2] ATLAS Collaboration, "Summary of the ATLAS experiment's sensitivity to supersymmetry after LHC Run 1 - interpreted in the phenomenological MSSM", JHEP **10** (2015), 134. `https://doi.org/10.48550/arXiv.1508.06608`.

[3] T. Šimko, L. Heinrich, H. Hirvonsalo, D. Kousidis, D. Rodríguez, "REANA: A system for reusable research data analyses", EPJ Web of Conferences 214, 06034 (2019), `https://doi.org/10.1051/epjconf/201921406034`.

[4] ATLAS Collaboration, "ATLAS Run 2 searches for electroweak production of supersymmetric particles interpreted within the pMSSM", ATLAS-CONF-2023-055 (13 Sep 2023). `https://cds.cern.ch/record/2870222`.