Volodymyr Kovalenko

Modernizing Unified: Resubmissions & Assignment of workflows.

1. Project Summary

1.1 Problem Statement

The Production and Reprocessing (P&R) team is responsible for central production tools that deliver simulation and data samples to analysis groups working for the Compact Muon Solenoid (CMS) experiment at the Large Hadron Collider (LHC). These samples are produced by millions of workflows that run every day on the Worldwide LHC Computing Grid, which consists of 1.5 million cores and 1.5 exabytes of storage. A key task for the P&R team is addressing the workflow failures across this grid, fixing them, and resubmitting the workflows out into the grid so that they can deliver the physics back to the researchers. Currently, the codebase that manages this important task, Unified, has grown out-of-date and complex, which complicates maintenance, makes it prone to bugs, and lacks flexibility for upgrades and integration with new features and technologies. This requires a lot of manual work and expert knowledge to update the system and has trouble with quick and effective troubleshooting, making it hard and slow to manage failures. Modernizing our tools will make them more reliable, maintainable , and effective, which will help thousands of workflows run more efficiently everyday across the LHC Computing Grid.

1.2 Solution

Our solution involves modernizing the key Unified modules crucial for managing workflow failures and resubmissions. We plan to overhaul the outdated codebase with a new, scalable architecture optimized for the Worldwide LHC Computing Grid. This will involve creating new classes to replace old modules like AutoACDC, Assignor, and Actor, ensuring seamless integration with modern technologies and leveraging already modernized utilities in our codebase. By adopting a Continuous Integration/Continuous Deployment (CI/CD) approach, we will enhance the software development process, facilitating more frequent and reliable updates. The modernization aims to boost reliability and streamline workflow management, making the system easier to upgrade and maintain. We will deploy these modules via Docker to ensure scalability and robustness across different environments. Additionally, robust unit testing and detailed documentation will enhance system functionality and maintainability, reducing the manual effort needed to handle workflow failures and ensuring more stable, efficient data delivery to CMS analysis groups.

1.3 Goals

Our goal this summer will be to transition the remaining Unified modules to the modern Unified branch. The modules that are left to be modernized are:

- AutoACDC
- Actor [only the functions singleRecovery, singleClone, the rest is WTC-related and can be deprecated]
- Assignor

- (if time allows) add to OpenSearch the outcome of ACDCs, begin putting together monitoring for ACDCs

Work has already begun on assignor. Many of these modules rely on utility scripts that have not all yet been modernized, though some have related dependencies, e.g. AutoACDC and assignor. As has happened with other modules that have been modernized already, it is sometimes preferable to refactor the responsibilities of each module, and perhaps even split them up across several new modules. This will be evaluated as we explore and detail the functionalities of each script.

2. Project Timeline

Week 1-2	● Set up the development environment. ● Familiarise with existing code documentation for AutoACDC, Actor, and Assignor. ● Outline merging strategy for modernization.
Week 3-4	● Develop new class for AutoACDC. ● Re-implement Actor functions: singleRecovery and singleClone. ● Begin preliminary merging activities.
Week 5-6	● Develop and test new Assignor module. ● Continue development and testing for AutoACDC. ● Execute branch merging as components are finalized.
Week 7-8	● Integrate and test new modules with existing utilities. ● Comprehensive branch merging.
Week 9-10	● Finalize development and system tests. ● Prepare all modules for deployment.
Week 11	● Deploy new modules via Docker. ● Complete documentation and final branch merging.