Impressed by Ingest – Efficient and Reliable Workflows

Juha Lehtonen, Kuisma Lehtonen, Aarno Tenhunen, Juha Törnroos, Ville-Pekka Vainio, Mikko Vatanen, CSC – IT Center for Science, firstname.lastname@csc.fi

We give an overview of our workflow solution, which is in production in our national digital preservation solution. Our preservation solution is designed to handle huge amount of data and hundreds of institutional customers. Given this, our solution must be able to handle at least the key processes automatically and survive temporal problems without human intervention. Further, our solution enables easy way to increase processing capacity when necessary. Our workflow solution is generic, and can be adopted to different kinds of repositories where efficient processes are needed.

Our national digital preservation repository, funded by the Ministry of Education and Culture within the Finnish Government, provides services for preserving the cultural heritage and research data. Currently our OAIS compliant digital preservation repository is in production having roughly a petabyte capacity. Finnish libraries, archives and museums can join as partners to our service, and they may submit their most valuable data to our solution for digital preservation. Eventually, our solution will provide national services to potentially hundreds of partners and is scalable to dozens of petabytes of preserved data. To achieve this efficiently we need our workflows generated for ingest, preservation and dissemination actions to be as fully automated, reliable, robust and error tolerant as possible. We believe that sharing our approach helps data repositories to develop reliable and scalable workflow systems for various purposes.

In our solution, each process is divided in tasks for the workflow. A task is a small and simple encapsulated part of the process, for example a checksum validation. The workflow system consists of a central scheduler and a set of workers, which execute tasks. The central scheduler is visible to all workers. It gives permissions to workers to execute tasks and knows, which tasks are in progress. Each worker can operate independently and in parallel in a distributed system.

In order to develop reliable and efficient repository, workflows should be as robust as possible. In case of temporal problems and for avoiding significant breaks in the production, the workflows must recover themselves automatically. All the tasks must be idempotent, which means, that a task must work the same way every time without causing any harmful side effects or needs for cleanup. In other words, a successful run of a task must give the same output with the same input every time it is executed. In a case of interruption, network error or any other problem, a task may fail. However, since the property of idempotence enables to execute the task again, a worker will automatically execute an interrupted task later. The error tolerance helps the system administration a lot and makes the workflow very reliable, since the administrator does not need to find out the current status of the workflow in abnormal situations.

Because of large amount of data and diverse hardware, the workflows must scale up to several parallel operations running in several servers. Currently our solution runs on five servers including 16 workers each, which sets the limit of parallel tasks to 80. Adding more servers with workers for task processing is easy, and it increases the processing efficiency. Using several servers requires a distributed filesystem. Having a distributed filesystem, all the workers in different servers can access to all needed information, for example to the output of another task. In our current implementation we use open-source software Luigi for the workflow and GlusterFS as the distributed filesystem.

Each task contains four parts: (1) *input*: values that are given for the task as parameters; (2) *prerequisites*: the results done by other tasks that are required to run the task; (3) *run*: execution of the task job itself; and (4) *output*: after a successful run, the task will write a PREMIS XML formatted result.

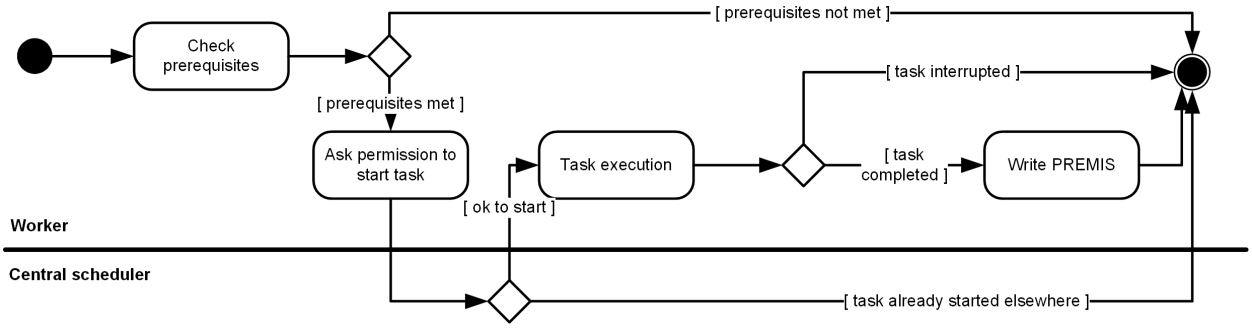

Figure 1: Task lifecycle in a workflow.

The task lifecycle is depicted in Figure 1. If a worker is free, it takes an uncompleted task and checks, if the prerequisites of the task are met. Then the worker requests permission from the central scheduler to run the task. Since the central scheduler has information of tasks in progress by workers, it is able to check if the task is already started elsewhere. The permission is given, if the task is not running elsewhere, and the worker runs the actual task. Finally the task will give a PREMIS XML report as an output, if the task completed successfully.

If the process consists of sequential tasks (i.e., a task can be run only if some another task is complete), the correct sequence is formed based on the prerequisites defined in the task. In this case, the worker checks whether the required results of the other tasks exist or not. The worker runs the task if all prerequisites are met. The task may do the actual job itself or run external software for it.

We require the tasks to be capsulated in favor of easy management. Since the metadata requirements, file formats, and other requirements are changing over time, our common national specifications must be updated on regular basis. In the workflow a task can be easily replaced with another task, and its prerequisites and PREMIS output can be changed according to the needs of the new task. These are all defined inside the task.

As an example, the data and the metadata can be submitted to our service as submission information packages (SIPs), which however, must be generated according to common national specifications. Since there are a large number of partners, we need to validate these SIPs automatically. These SIPs are then validated automatically in the ingest process. Single validation process includes several tasks, for example uncompressing, virus check, technical structure validation, digital signature validation, metadata validation and file format validation of digital objects. In some cases this workflow requires a sequence (for example, a SIP needs to be uncompressed before the actual validation), but some parts can be done in parallel (for example, the validation of different SIPs or validation of digital objects in a SIP). When all SIP validation tasks are completed for a SIP, and the SIP validates through the process, the workflow continues to AIP generation and storing. Eventually, the workflow gives ingest reports for the producer, one report for each SIP.

We presented our generic and flexible workflow solution for handling data in digital preservation where scalability, efficiency and robustness are key elements. Our workflow solution has an idempotence property. This property enables constructing very reliable workflows enabling large repositories processing huge amount of data as automatically as possible. Workflow is in production in our national digital preservation solution for cultural heritage material and research data.