Workflow Study in PanDA WMS

FaHui Lin (UTA)
on behalf of PanDA team
250130

Existing Components Supporting Workflows

- **JEDI**:
  - With task definitions submits individual jobs to PanDA and keeps track of job progress and handle retries
  - JEDI task usually as the basic step of workflows
- **DEfT / ProdSys2**: manage production tasks, chains, bags; DC
  - Formulates the tasks, chains of tasks and also bags of tasks or bags of chains with necessary parameters, and provides higher-level task monitoring
  - Interfaces data management services in order to manage datasets for aggregating data generated by tasks
  - DC (data carousel) core machinery: Create DDM rules to stage datasets, control requests to respect tape profiles, etc.
- **iDDS** : use cases of special workflows (DC, HPO, AL), pchain workflows
  - Empowers PanDA WMS with extra workflows (use cases) by intelligently transform and deliver data to the processing workflow in fine-grained approach
  - Special workflows (non-exhaustive):
    - DC (data carousel): poll DDM rules; once files are staged, trigger JEDI to submit job
    - HPO (hyperparameter optimization): PanDA jobs run training with hyperparameters ; iDDS generates new sets of hyperparameters according to results of last cycle of PanDA jobs to improve training
    - AL (active learning): run learning tasks in iDDS internal batch-system to analyse the outputs of the PanDA task, whose results decides whether to generate new PanDA tasks or to terminate
  - pchain workflows: arbitrary workflow defined by users in WDL (workflow description language, e.g. Snakemake, CWL); iDDS runs as workflow engine to finish the workflow (submit PanDA tasks as workflow steps)

To Move On

- Goal of S4 workflow: Make PanDA WMS an interactive and dynamic workflow-oriented platform
  - Expand support of complex workflows for both production and analysis
  - Optimize algorithms in the system with the awareness of entire workflows
    - Currently PanDA/JEDI are not aware of the entire workflow to optimize the processing (e.g. task brokerage is sub-optimal without knowledge on relationship among parent and child tasks of pchain)
  - Cleaner interface for users about workflows
    - Currently lack of GUI for users to manage complex workflows
- However, it's challenging to decide the best direction to implement
  - There are many concerns (Which functionalities to run in PanDA/JEDI and which to run on iDDS (technical, practical, political concerns), ideas to be agreed by all PanDA, DEfT and iDDS developers, possible overlapping/duplicate functionalities in other components, may increase too much complexity in PanDA, quick sloppy vs carefully-designed implementation, ...)
  - Every once in a while various ideas have been proposed and discussed internally. Some were rejected

Rejected Ideas

- Merge iDDS & PanDA database tables
  - To share information about workflows among iDDS and PanDA/JEDI
  - Reasons for rejection:
    - For the same object (e.g. dataset), iDDS and JEDI store them respectively in their own tables. The states/timestamps of the object in iDDS table are only used in iDDS, and vice versa
    - Shared read-only tables are technically possible

- In PanDA world, unify pchain workflow and iDDS special workflows
  - To introduce “workflow objects” in PanDA to interface iDDS
    - **Normal pchain workflow**: arbitrary workflow defined by end users
    - **Co-workflow**: internal workflow to interface iDDS (or Rucio, other components) associated with special tasks which requires iDDS special workflow functionalities (DC, HPO, AL, etc.)
  - No workflow processor in PanDA/JEDI; only add workflow objects to interface iDDS (the workflow processor)
  - Reasons for rejection: Will over-complexify PanDA system; more direct approaches have been accepted
    - Difficult and unnatural to design common workflow object to represent both types of workflows
    - Data Carousel is not really “workflow” from user’s point of view, but an internal feature of WMS
    - For DC, now just plan to implement core machinery in PanDA/JEDI. No need to be handled with new workflow objects
    - For HPO, AL, etc. the co-workflows are also internal and not visible by users (less meaningful for monitoring)

Current Plans

- Data Carousel implementation in PanDA/JEDI for analysis
  - Implement Data Carousel machinery in PanDA/JEDI with new PanDA DB tables
    - In consultation with Misha - developer of production DC on Prodsys2
  - Basic machinery done. Ongoing to be enriched with functionalities and to extend to production
- Introduce pchain workflow as first-class entity (like jobs, tasks) in PanDA/JEDI
  - Only for pchain workflow (arbitrary via WDL)
  - Implement workflow processor in PanDA/JEDI + new PanDA DB tables for workflows & steps
    - In consultation with Wen - developer of iDDS & various workflows
    - Having experience from DC in PanDA/JEDI to migrate core features from other component to PanDA/JEDI
  - Improve existing algorithms in PanDA/JEDI with consideration of entire workflows
  - GUI on new monitoring (helps from Tanya)
- No plan to merge/integrate other existing workflow use-cases into PanDA/JEDI
  - Prodsys2 is optimized for production tasks, chains and bags
  - Special use cases (HPO, AL, workflows of non-ATLAS experiments) are well supported by iDDS
- Approaches are opted to change in the future
  - According to new needs or concerns

Backup

Idea about Merging DBs of iDDS & PanDA

- Tables in iDDS:
  - `REQUESTS`: requests to iDDS; request is converted to workflow
  - `COLLECTIONS` and `CONTENTS`: datasets and files
  - `TRANSFORMS`: transforms (works or steps) broken down from a workflow. A transform may map to a JEDI task
  - Others (processings, events, maps, archives, ...)
- Tables in PanDA and iDDS to merge or share ?
  - Although some PanDA tables and iDDS tables have some duplicate information, I prefer they stay separate
    - For the same object, each table stores states/timestamps of it used in each of iDDS or JEDI respectively
    - For monitoring, we need to plan which data are relevant to show to users
      (From iDDS table? From PanDA table? Merge of both?)
  - Maybe okay to share tables (some, all) with each other
    - REQUESTS, COLLECTIONS, CONTENTS - for JEDI to known the inputs/outputs of the entire workflow
  - Possibilities of DB schema & table synonym
    - Oracle: use SYNONYM (or VIEW ?)
    - Postgres: use search_path or VIEW
      - `ALTER DATABASE DOMA_PANDA SET search_path = DOMA_IDDS ;`
      - `CREATE VIEW DOMA_IDDS.WORKFLOW_TABLE AS SELECT * FROM DOMA_PANDA.WORKFLOW_TABLE;`
    - One connection to access different schemas

Example:

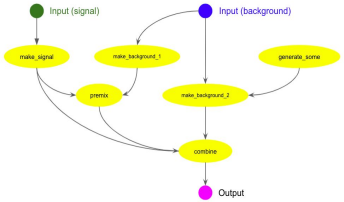
The same dataset information (output of a workflow) in iDDS collections table and PanDA JEDI datasets table

The column of states (states, timestamps, locks, counters) are only meaningful in each components (iDDS or JEDI) respectively

```
SQL> SELECT * FROM ATLAS_IDDS.COLLECTIONS WHERE request_id=726799 AND coll_id=1256905;

   COLL_ID |  COLL_TYPE | TRANSFORM_ID | REQUEST_ID | WORKLOAD_ID | RELATION_TYPE | SCOPE
   1256905 |          1 |       617242 |     726799 |    39858747 |             1 | user.flin

 NAME
user.flin.test_wf_.simple.0_001_bottom_results.root.

     BYTES |     STATUS |  SUBSTATUS |    LOCKING | TOTAL_FILES | STORAGE_ID |  NEW_FILES | PROCESSED_FILES | PROCESSING_FILES |
         0 |          4 |          4 |          0 |           0 |            |          0 |               0 |                0

  | PROCESSING_ID |    RETRIES | CREATED_AT          | UPDATED_AT          | NEXT_POLL_AT        | ACCESSED_AT
  |               |          0 | 2024-06-12 10:41:39 | 2024-06-12 10:50:05 | 2024-06-12 10:41:39 | 2024-06-12 10:50:05

EXPIRED_AT          | COLL_METADATA                                                                | FAILED_FILES
                    | {}                                                                           |            0

  | MISSING_FILES |  EXT_FILES | PROCESSED_EXT_FILES | FAILED_EXT_FILES | MISSING_EXT_FILES
  |             0 |          0 |                   0 |                0 |                 0
```

```
SQL> SELECT * FROM ATLAS_PANDA.Jedi_Datasets WHERE jeditaskid=39858747 AND type='output';

JEDITASKID |  DATASETID | DATASETNAME
  39858747 |  550676347 | user.flin.test_wf_.simple.0_001_bottom_results.root.550676347.550676347

TYPE                | CREATIONTIME        | MODIFICATIONTIME    | VO              | CLOUD      | SITE
output              | 2024-06-12 10:47:49 | 2024-06-12 10:49:33 | atlas           |            | BNL/SCORE

|   MASTERID | PROVENANCEID | CONTAINERNAME
|            |              | user.flin.test_wf_.simple.0_001_bottom_results.root/

STATUS              | STATE               | STATECHECKTIME      | STATECHECKEXPIRATIO | FROZENTIME          |     NFILES
done                | closed              | 2024-06-12 10:49:33 |                     |                     |          1

| NFILESTOBEUSED | NFILESUSED | NFILESONHOLD |    NEVENTS | NEVENTSTOBEUSED | NEVENTSUSED | LOCKEDBY | LOCKEDTIME
|                |          0 |            0 |          0 |               0 |           0 |          |

| NFILESFINISHED | NFILESFAILED | ATTRIBUTES                                          | STREAMNAME | STORAGETOKEN
|              1 |            0 |                                                     | OUTPUT0    |

| DESTINATION                                          | TEMPLATEID | NFILESWAITING | NFILESMISSING
| BNL/SCORE                                            |  550675154 |             0 |             0
```

Custom pchain Workflow



pchain workflow

parsed into

PanDA Workflow

Step 1

Step 2

Step 3

Workflow and steps registered in PanDA DB

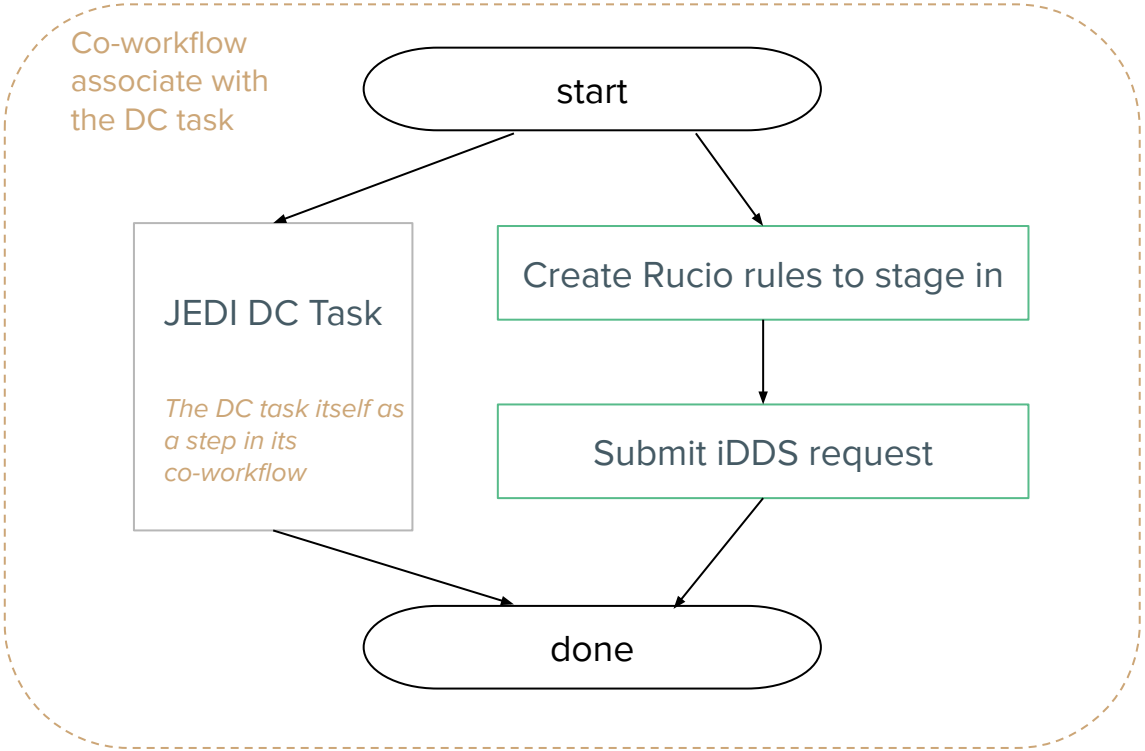
Generate iDDS's object
Submit request to iDDS

iDDS

Process steps
Submit tasks

Update back info of tasks to PanDA steps

Data Carousel Co-workflow



Other special tasks (like HPO, AL) will also have co-workflows (including iDDS request)

Concerns about Workflow / Co-workflow

- pchain workflows & co-workflows are quite different
  - Call them both “workflow objects” because they may both interface iDDS, but they are different:
    - pchain workflow:
      - User-defined, can be (want to support) complicated DAG, AG (with loop, conditions, etc.)
      - Steps are (mostly) JEDI tasks submitted by iDDS
      - Want brokerage to consider entire custom workflows
    - Co-workflow:
      - Often just simple, linear, per-defined workflows (e.g. DC)
      - Steps require JEDI/PanDA agent to process them (if not wanting to run them with iDDS transformations)
      - Nothing to do with brokerage
- Hard to design common workflow object to represent both