A FAIR-oriented Data-lake Workflow: Take One

Marek Szuba / GSI

WP2 Fortnightly Meeting, 2020-10-21

Overview

- FAIR computing models still in early stages of development
- Work flows and storage requirements vary between experiments
  - but: primarily ROOT-based
- Primary assumptions, driven by CBM:
  - petabytes of raw and Monte-Carlo data per year
  - individual files in the order of 1 GB
  - regular reprocessing, no storage of AOD files
  - nearly real-time processing rates
- For the rehearsal: *placeholder ROOT data analysis*
  - ROOT tutorial `h1analysis`, patched to write an output file
  - input files available at `https://root.cern.ch/files/h1/`

Environment

- Architecture: primarily amd64
- OS: Linux (several distros) with Python3
  - batch jobs run in Singularity containers
- DL software:
  - voms-clients (Java version)
  - rucio-clients from PyPI
  - gfal2-bindings from CERN Gitlab
- Authentication: X.509 proxy certificates
  - for containers, injected through a shared directory — no private keys in the image itself
  - renewal now: periodically run shell script calling `voms-proxy-init --noregen`
  - in the future: likely MyProxy
- Storage QoS classes: `CHEAP-ANALYSIS` for input data, `SAFE` for analysis results, `OPPORTUNISTIC` for log files

Container Image

- Base image: `docker://rootproject/root:latest` (Ubuntu-based)
  - quicker and simpler than adding ROOT to the Rucio image
- Extended under Docker, exported and converted to SIF
  - personal preference — could be done directly in Singularity
- Summary of changes:
  1. Install `voms-clients-java`, `python3-pip`, `gfal2`, `gfal2-plugin-file`, `gfal2-plugin-gridftp`, `gfal2-plugin-http`, `libgfal2-dev` with Apt
  2. Install `rucio-clients` with Pip3
  3. Install `gfal2-bindings` manually, as per ESCAPE tutorials
  4. Deploy Grid CA certificates/CRLs and ESCAPE VOMS configuration
  5. Deploy minimal rucio.cfg

Rucio Configuration

rucio.cfg:

```
[client]
rucio_host = https://escape-rucio.cern.ch:32300/
auth_host = https://escape-rucio.cern.ch:32301/
ca_cert = /etc/grid-security/certificates/CERN-bundle.pem
auth_type = x509_proxy
```

Environment:

- `RUCIO_ACCOUNT` — Rucio user name
- `X509_USER_PROXY` — path to proxy certificate

Input Data

- During earlier testing: 1 GiB-long random files, uploaded to GSI-ROOT and replicated to QoS classes
- For the work-flow test, actual ROOT files:
  - 1 data set of 4 files
  - about 300 MiB in total
  - uploaded to GSI-ROOT, then replicated in 2 copies to QOS=CHEAP-ANALYSIS
  - original upload rule deleted once replication succeeded

```
readonly home_rse='GSI-ROOT'
readonly scope='FAIR_GSI_SZUBA'
rucio upload --scope ${scope} --rse ${home_rse} ${scope}:root-cern-ch-files-h1 dstar*.root
source wait_for_replication.sh
wait_for_replication `rucio add-rule ${scope}:root-cern-ch-files-h1 2 QOS=CHEAP-ANALYSIS`
upload_rule=`rucio list-rules ${scope}:root-cern-ch-files-h1 | grep " ${home_rse} " | cut -f1 -d' '`
rucio delete-rule ${upload_rule}
```

wait_for_replication.sh

```
wait_for_replication() {
    if [[ ${#} -lt 1 ]]; then
        echo "Usage: ${FUNCNAME} replication_rule_id"
        return -1
    fi
    local replication_rule="${1}"

    local replication_state
    while :; do
        replication_state=$(rucio rule-info "${replication_rule}" | \
                grep '^State:' | sed -e 's/^State:\s\+//')
        case "${replication_state}" in
            OK)
                echo "Done!"
                return 0
                ;;
            REPLICATING)
                ;;
            STUCK)
                echo "Replication has become stuck!"
                return 1
                ;;
            *)
                echo "Unknown replication state '${replication_state}'!"
                return -1
                ;;
        esac

        sleep 1m
    done
}
```

“Analysis” Job

```
readonly home_rse='GSI-ROOT'
readonly scope='FAIR_GSI_SZUBA'
readonly jobid="h1_$(date -u +'%Y%m%d_%H%M%S')"

rucio download ${scope}:root-cern-ch-files-h1

export H1="$PWD/root-cern-ch-files-h1"
root -q -b "$ROOTSYS/tutorials/tree/run_h1analysis.C" 2>&1 | tee OUT

mv h1analysis.root "${jobid}.root"
mv OUT "${jobid}.log"
rucio upload --scope ${scope} --rse ${home_rse} "${jobid}.root"
rucio upload --scope ${scope} --rse ${home_rse} "${jobid}.log"
rucio add-rule "${scope}:${jobid}.root" 1 QOS=SAFE
rucio add-rule "${scope}:${jobid}.log" --lifetime 2592000 1 QOS=OPPORTUNISTIC
# remove the home_rse replication rules once QOS replication has concluded
rucio add-dataset "${scope}:${jobid}"
rucio attach "${scope}:${jobid}" "${scope}:${jobid}.root" "${scope}:${jobid}.log"

rm -r "${jobid}.root" "${jobid}.log" root-cern-ch-files-h1/
```

Results and Conclusions

- Rules and replication generally work fine nowadays
- Upload procedure somewhat tedious:
  - cannot upload directly to QOS, must use a “home” RSE
  - unlike uploads, replication offers little feedback on status
  - even now, replication time quite variable
- Out-of-band “replication watchdog” might be required
- Is it safe to remove home-RSE rules before QOS ones succeed?