RODS®

iRODS Policy: Read-only local analysis staging policy for BRAIN-I

Terrell Russell, Ph.D. @terrellrussell Chief Technologist, iRODS Consortium June 8-11, 2021 iRODS User Group Meeting 2021 Virtual Event

Collaboration between

• RENCI

- Steve Cox, Director, Software Architecture
- Terrell Russell, Chief Technologist, iRODS Consortium
- UNC Neuroscience Microscopy Core (NMC) in the UNC-Chapel Hill School of Medicine
 - Michelle Itano, Director
- Stein Lab in the UNC-Chapel Hill Department of Genetics
 - Jason Stein, Director
 - Oleh Krupa, Ph.D.

The Stein Lab studies how variations in the genome affect the structure and development of the brain, and in doing so, create risk for neuropsychiatric illnesses. One of the lab's research projects involves using high-powered optical microscopes to create extremely detailed images of mouse brains.

BRAIN-I Project - Architecture

iRODS



Developed with stakeholders:

- scientists doing the capture
- lab technicians
- lab director
- microscopy core director
- RENCI infrastructure
- iRODS developer / administrator

Over the course of 2-3 months.

Four main use cases defined and agreed to...

NMC1. Import of data to iRODS: A lab member saves image data into a well-known location on a local disk attached to the iRODS server (Maybe a folder named 'iRODS import'). (Maybe this can be automated later, but for now... just 'saved/moved' is good enough). The lab member expects the data to appear in iRODS, perhaps with some particular permissions, perhaps with some particular metadata (provided or extracted or associated). The lab expects the data to be managed/moved off the local disk, so it doesn't fill up. Ideally, once in iRODS, there would be a backup made to a second more secure location. Once copied into another iRODS location with more storage, the import files are removed from the local disk.

SOLUTION - Storage Tiering (and possibly automated ingest)

NMC2. Import of data to iRODS and local analysis: A lab member saves image data into a wellknown location on a local disk attached to the iRODS server (Maybe a folder named 'iRODS import'). (Maybe this can be automated later, but for now... just 'saved/moved' is good enough). The lab member expects the data to appear in iRODS, perhaps with some particular permissions, perhaps with some particular metadata (provided or extracted or associated). The lab expects the data to be managed/moved off the local disk, so it doesn't fill up. Ideally, once in iRODS, there would be a backup made to a second more secure location. Once copied into another iRODS location with more storage, the import files are moved to an 'analysis' folder on the local iRODS server.

SOLUTION - Storage Tiering (and labels for 'local analysis') local analysis could be read-only in the vault, perhaps with permissions changed(?!), or another local copy if read/write was important. And any products of local analysis would get routed back through the front door of iRODS. Also possible NFSRODS, but concerns about network latency are real.

NMC3. Local iRODS analysis: Analysis of files on the local iRODS server from the 'analysis' folder using the local hardware accessible. Could be a custom python script, Matlab code, or through iRODS on ImageJ or Napari. Ideally, would utilize as much computational power as is available, without making it impossible for another user to run a job. Not sure if this can be set to somehow utilize up to 100% or 90% power, but then if another job is started to drop down to maybe 75% or something like that? Or maybe only do that if the job is set to last more than 6 hours, or some relatively arbitrary 'long time'. Leaving 25% for other jobs to still run during that time period? Otherwise for 'shorter' times jobs would be run sequentially with full computational power? Would ideally save analyzed files into the 'iRODS import' folder where again it would be copied and moved to a larger external iRODS server, and removed from the local drive to free up space. Files remaining in the 'analysis' folder would still be accessible to run code on locally.

SOLUTION - taken care of by solution to NMC2

NMC4. External user iRODS access to published data: Published data would be put on an iRODS protected folder and available post publication for download. Ideally, this would be a very quick process, not requiring a person to verify the request validity. But would also be good to mark how many times files had been downloaded, track who downloaded (maybe just by email address and verifying that address was real?), and if possible allow users to only download specific file sets and not the entire file set. Would want to also ensure that downloading this data wouldn't take up all the bandwidth for the iRODS server.

SOLUTION - storage tiering and labeling of particular collections or data objects - policy can fire and run the 'publish()' function... whatever that is defined to be...



- Automatic Storage Tiering to primary storage at RENCI
- Manual targeting of files to NMC for local analysis
- Manual targeting of files to future location as published

- 3 storage resources
 - NMC-ingest on workstation
 - primary at RENCI
 - NMC-analysis on workstation
- automated ingest configured to register in place into NMC-ingest
- storage tiering configured
 - NMC-ingest on workstation -> primary at RENCI
- local NMC analysis use case
 - pep_metadata_add_post()
 - replicate collection or object to NMC-analysis
 - $\circ~$ set physical permissions to read-only for others on NMC workstation
 - pep_metadata_remove_post()
 - trim replica from NMC-analysis
- publishing use case
 - pep_metadata_add_post()
 - replicate/checksum/doi to public area (TBD)

- Always a replica in primary storage at RENCI after initial migration process
 - could be from nearly 'real-time' to '1 hour' to '1 day' or '1 week'
 - set minimum restage tier to RENCI-primary, so nothing ever restages
- Allows for manual replication/tagging to manage analysis area (a second replica)
- Allows for future manual tagging for publication
- Cloud Apps pull from always-available primary at RENCI, do not trigger replication

As part of the BRAIN-I project, this iRODS policy set defines the policies for data analysis, replication, and retention in the NMC.

There are two parts of the policy managing the data flow within the iRODS Zone:

• Automatic

The iRODS Storage Tiering Framework is handling newly ingested data and moving it into the long-term storage housed at RENCI. RENCI is providing storage and visualization tooling that prioritizes that local, long-term storage.

• Manual

When NMC staff want to run local analysis on data already in the iRODS namespace, they can 'tag' the data of interest, and this policy will manage the replication to their local machine, set permissions, and prevent removal of that data from the system until it has been 'untagged'. Once 'untagged', the data will be trimmed from the researchers' local storage and remain housed only in long-term storage at RENCI.

Storage Tiering Efficient Policy-Driven Storage Utilization

iRODS CAPABILITY



BRAIN-I Project - Architecture

iRODS





\$ git clone https://github.com/bats-core/bats-core \$ time bash bats-core/bin/bats test nmc analysis.bats ✓ tag a collection ✓ tag a data object ✓ untag a collection ✓ untag a data object ✓ overwrite a tagged data object ✓ overwrite a data object under a tagged collection ✓ trim a tagged data object - DISALLOWED ✓ trim a data object under a tagged collection - DISALLOWED ✓ remove a tagged data object - DISALLOWED ✓ remove a tagged collection - DISALLOWED ✓ remove a data object under a tagged collection - DISALLOWED ✓ remove a collection under a tagged collection - DISALLOWED ✓ remove a collection containing a tagged data object - DISALLOWED ✓ remove a collection containing a tagged collection - DISALLOWED ✓ untag an enqueued data object - DISALLOWED ✓ untag a collection with an enqueued descendent data object - DISALLOWED

16 tests, 0 failures

real 2m4.745s user 0m8.606s sys 0m2.172s

```
$ iquest "%s" "select count(DATA_ID) where RESC_NAME = 'nmc-ingest'"
0
```

\$ iquest "%s" "select count(DATA_ID) where RESC_NAME = 'nmc-analysis'"
0

\$ iquest "%s" "select count(DATA_ID) where RESC_NAME = 'renciResc'"
921670

irods

- Automated Ingest (Landing Zone)
- Manual tagging for publication
- Additional microscopes
- Additional labs

irods



Code:

https://github.com/irods/irods_policy_examples/tree/master/nmc_analysis

Configuration:

```
nmc_target_resource = 'nmc-analysis'
nmc_remote_hostname = 'localhost'
nmc_change_permission_script = 'nmc_set_permissions.sh'
nmc_a = 'nmc'
nmc_v = 'analysis'
nmc_u = ''
nmc enqueued = '{}::enqueued'.format(nmc a)
```



Questions?