

iRODS

Hard Links Rule Engine Plugin

Kory Draughn
Software Developer
iRODS Consortium

June 9-12, 2020
iRODS User Group Meeting 2020
Virtual Event

Hard Links - Why?

https://github.com/irods/irods_client_nfsrods/issues/57



trel commented on Oct 16, 2019

Member



When putting a file or overwriting a file via WinSCP...

From NFSRODS log:

```
2019-09-16 15:18:58.917 ERROR [NFSServerV41] - Unhandled exception:  
java.lang.UnsupportedOperationException: Not supported  
    at org.irods.nfsrods.vfs.IRODSVirtualFileSystem.link(IRODSVirtualFileSystem.java:  
    at org.dcache.nfs.vfs.PseudoFs.link(PseudoFs.java:202)  
    at org.dcache.nfs.v4.OperationLINK.process(OperationLINK.java:68)  
    at org.dcache.nfs.v4.NFSServerV41.NFSPROC4_COMPOUND_4(NFSServerV41.java:204)
```

Might be possible via GenQuery to ask how many logical paths are pointing to the same physical path on the same resource... If > 1, then do something smart...

- Applications such as WinSCP implement some operations using hard links.
- Provide a generic solution for the community and future projects (e.g. SMBRODS).

What is a hard link?

In Linux, all file information is stored in an i-node.

A hard link is the mapping between a filename and an i-node.

i-nodes can have multiple hard links.

Hard Link representation in iRODS

A hard link has the following characteristics:

- All affiliated data objects have different logical paths
- All affiliated data objects point to the same replica
- All affiliated data objects have the following AVU:
 - name: **irods::hard_link**
 - value: **<UUID>**
 - unit: **<resource_id>**

The (UUID, resource_id) tuple uniquely identifies a hard linked replica.

A data object can be a member of multiple hard link groups.

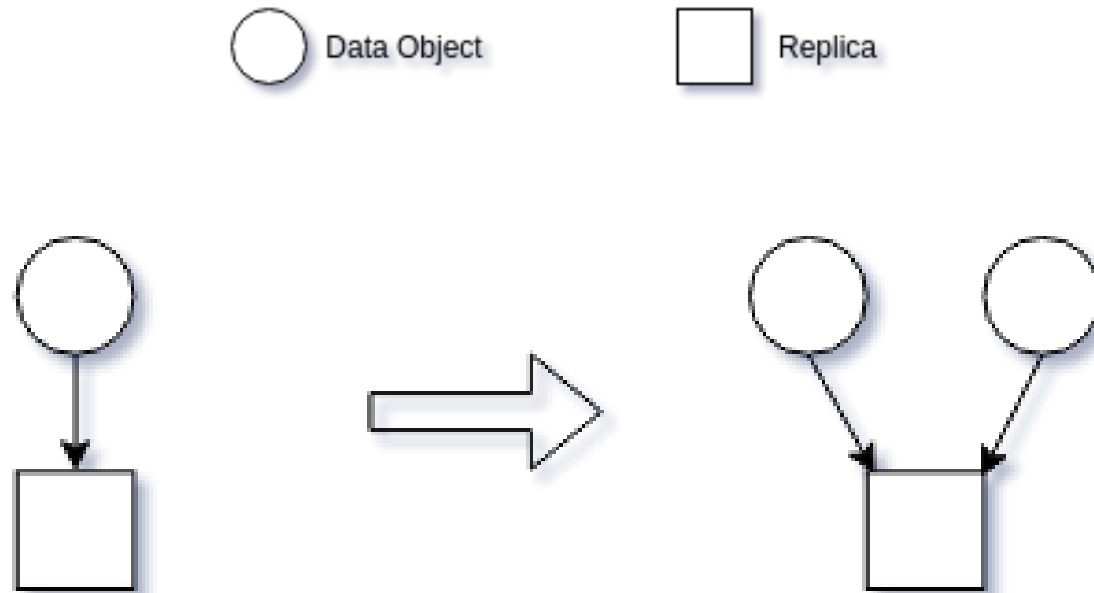
Policy Enforcement Points

- pep_api_data_obj_rename_pre
- pep_api_data_obj_trim_pre
- pep_api_data_obj_unlink_pre
- pep_api_data_obj_phymv_post

Maintaining the behavior of these operations required changes to the Rule Engine Plugin Framework (REPF).

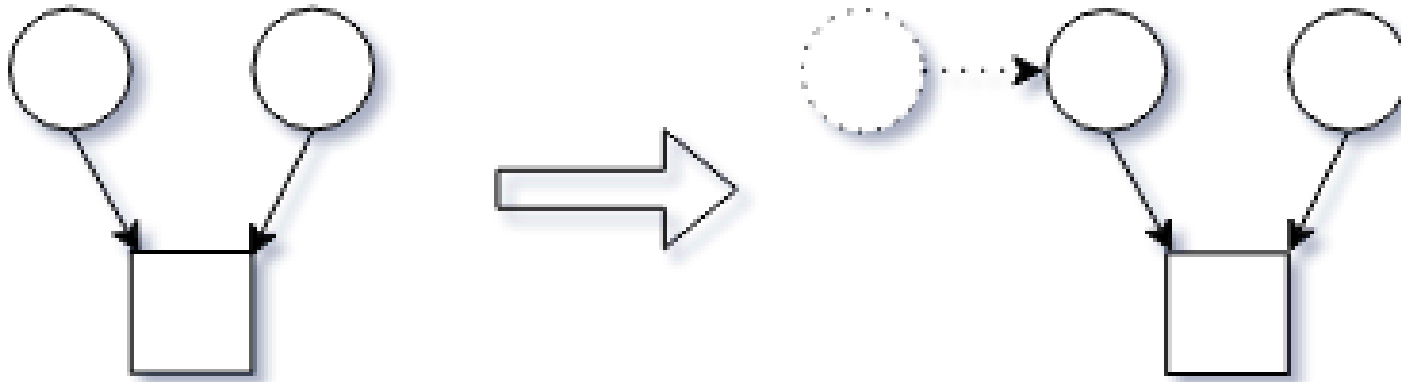
- RULE_ENGINE_SKIP_OPERATION
 - Instructs the REPF to skip everything following the pre-PEP
 - Allows PEPs to override operations
 - "finally" PEPs are always triggered

Example: Creating a Hard Link



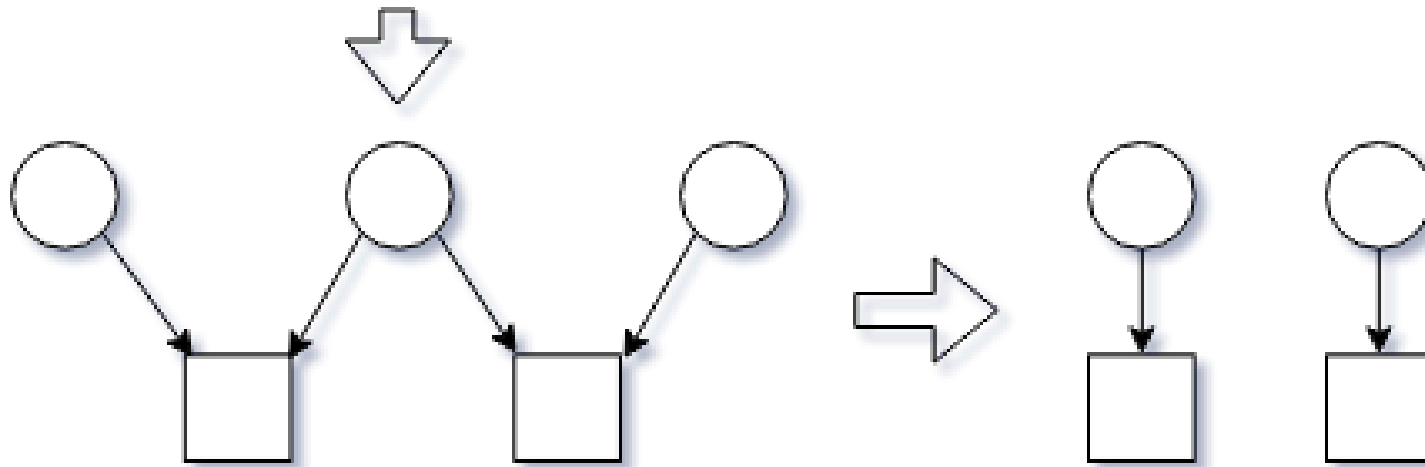
- Generates new metadata and attaches it to each data object
- Or, attaches the existing metadata to the newly registered data object
- Copies all permissions from the source data object to the newly registered data object

Example: Renaming a Hard Link



- Renaming or moving a hard linked data object only updates the logical path of that data object
- The physical path is never touched

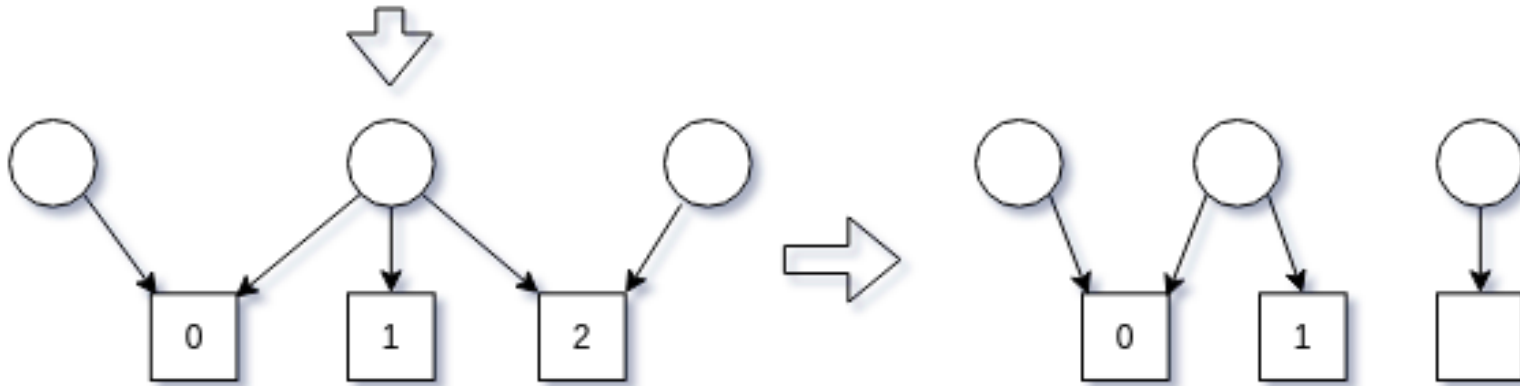
Example: Unlinking a Hard Link



- The data object being unlinked is a member of two hard link groups
- The targeted data object is unregistered and all metadata is removed from the remaining hard link members
- The result is two data objects without any hard link metadata

Example: Trimming a Hard Link

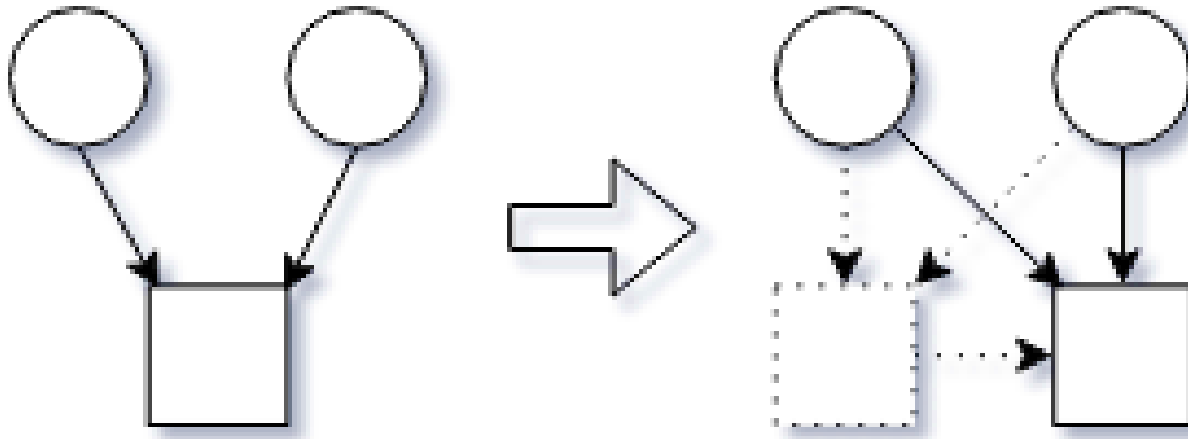
```
$ itrtrim -n2
```



- Trimming a hard linked data object never unlinks it
- Replica two is unregistered from the targeted data object
- The metadata shared by the data objects pointing to replica two is removed from both data objects.

Example: Physically moving a hard linked replica

```
$ iphymv -S <src_resource> -R <dst_resource>
```



- The units of the hard link metadata is updated to reference the new resource id
- All data objects in the hard link group are updated to reflect the new location of the replica

Important things to know

- Never modify the hard link metadata directly
- Hard links do not track the first data object which referenced the replica
- "post" PEPs are never triggered when handling a hard linked data object
- "finally" PEPs are always triggered

Apparently, this is hard 😅!

Since release, we've identified that the plugin needs some additional work to make it play well with others.

Some other rule engine plugins depend on the "post" PEP to operate properly.

Questions?

Hard Links Resources:

- Repo: https://github.com/irods/irods_rule_engine_plugin_hard_links
- RFC: https://github.com/irods/irods_rfcs/blob/master/0005_hard_links.md