

Updates/Improvements for gb2_ds_rm

BIIDCD-1448

BNL-UM Computing Meeting

Justin Guilliams 9/1/21



THE UNIVERSITY *of*
MISSISSIPPI

Updates for gb2_ds_rm

- Improvements for tool for next release:
 - (1.) Single-file Deletion
 - (2.) Empty-dataset Deletion
- PR #492 (<https://stash.desy.de/projects/B2DC/repos/belledirac/pull-requests/492/overview>)
- Due to comments from Ueda-san and Miyake-san, two new methods written to be used for gb2_ds_rm
- Updated hackMD page: <https://hackmd.io/2koQkkltRzqyIDUWet6N1Q?view>

Two new methods

1. `listDirectory`
2. `removeFileAsync`

listDirectory

- The layers for code for the gb2 tools are:

gb2 tool -> Controller -> Manager -> FileCatalog

- Originally, I was calling a method from FileCatalog in rmAsync in datasetCLController for when an empty dataset is being deleted - This is a no no (apparently 😊)
- To abide by these rules, a new method in Manager was needed:

```
def listDirectory(self, lpns):  
    """  
    Calls the listDirectory method from FileCatalog and returns the results of that method.  
  
    :param lpns: List of LPNs  
    :return: S_OK(directory contents for supplied list of LPNs) or S_ERROR(message)  
    """  
    return self.__fileCatalog.listDirectory(lpns)
```

removeFileAsync

- Comment on PR from Ueda-san:

Ueda Ikuo

Shouldn't we remove files asynchronously, using Rucio?

0) In principle, all the deletion should be treated by Rucio. (and this is `rmAsync`)

1. deleting files can fail, or can take time if the list is long, and also files on banned SE cannot be deleted, and the users need to come back later to delete them all.
2. `DataManager.removeFile` first deletes the replicas, and then the file entry from the catalog. On the other hand, `RucioFileCatalog` does nothing when `removeReplica` is called, and does the work when `removeFile` is called, which is not expected in `DataManager.removeFile`. We better follow the principle (0).

- The original suggestion was to re-write `manager.removeFile`, but other methods use this method
- A new method was written in `Manager` to handle deletion of files:

```
def removeFileAsync(self, lfn):  
    """  
    Removes files asynchronously using the removeFile method from FileCatalog.  
  
    :param lfn: Supplied file path  
    :return: S_OK(lfn) or S_ERROR('Message')  
    """
```

Updates for gb2_ds_rm

- Additionally, a bug was found in the current implementation of `manager.removeFile` that incorrectly searched the return values from `FileCatalog.exists`
 - This bug was fixed and is included in the commit
- Commit with new methods made - no further comments made yet

