

Nils Haustein

IBM European Storage Competence Center
Am Weiher 24, 65451 Kelsterbach
nils_haustein@de.ibm.com

Mauro Tridici

Euro-Mediterranean Center on Climate Change
via Augusto Imperatore 16, 73100 Lecce
mauro.tridici@cmcc.it



Integration of iRODS with IBM Spectrum Archive™ Enterprise Edition

A flexible tiered storage archiving solution

iRODS UGM 2020, June 9-12, 2020, Tucson, Arizona, USA

[Authors retain copyright. Nils Haustein – IBM European Storage Competence Center, Mauro Tridici – Euro-Mediterranean Center on Climate Change]

Contents

- Introduction.....3
- iRODS.....3
- Tiered Storage File System.....4
- Solution Integrating iRODS with IBM Spectrum Archive.....4
 - Prevent transparent recall.....5
 - Display file status.....6
 - Set quota for the entire file space.....6
 - Extracting and ingesting metadata.....6
- References.....8
- Disclaimer.....9

INTRODUCTION

A tiered storage system provides lower total cost of ownership for large volumes of data by storing data on the most appropriate storage tier (flash, disk and tape). Independent studies have shown that total cost of ownership of tape solution provides an expected TCO that is more than 80% lower than that of the all-disk solution [1].

While tape storage is suitable for storing large volumes of data over long periods of time at lower cost, access time to data on tape is significantly higher than to data on disk. Providing data from tiered storage file systems with tape in multi-user environment bears several challenges. These challenges and solutions are further elaborated in this blog article [2].

In summary, tiered storage file systems with tape storage are a blessing and a curse. The blessing is that the user can see all files regardless if these are stored on disk or tape. Cursing starts when the when the user opens a file that is stored on tape because the recall takes one or more minutes. Unfortunately, the user is not aware that the file is on tape because standard file systems do not indicate whether the file is on disk or on tape. It gets even worse if the many users simultaneously open several files that are on tapes. This causes even longer waiting times because transparent recalls are not tape optimized.

To address these challenges, the user must be able to determine the location of files and request files from tapes to be recalls. These recall requests coming from multiple users can be queued and recalled periodically in a tape optimized manner whereby the files are sorted by the tape-ID and the location on tape. The combination of iRODS with IBM Spectrum Archive Enterprise Edition can accommodate this.

In this paper Mauro Tridici from the Euro-Mediterranean Center on Climate Change (CMCC) and Nils Haustein from the IBM European Storage Competence Center give a brief introduction to iRODS and explain examples for integrating iRODS with IBM Spectrum Archive and its advantages. For more information refer to our whitepaper [3].

IRODS

iRODS software is a data management layer - maintained by the iRODS consortium - that sits above the storage that contain data, and below domain-specific applications [4]. The data virtualization capabilities of iRODS make it a one-stop shop for all data regardless of the heterogeneity of storage devices. Whether data is stored on a local hard drive, on remote file systems or object storage, iRODS' virtualization layer presents data resources in the classic files and folders format, within a single namespace.

iRODS is open-source, data management middleware that enables users to:

- Access, manage, and share data across any type or number of storage systems through iRODS APIs (iCommands, REST, WebDAV, Python, C++, Java)
- Automate workflows through powerful rules and microservices
- Search and find data through descriptive metadata and query tools

iRODS rules are executed based on conditions or, in iRODS terminology, Policy Enforcement Points (PEPs). iRODS can be integrated with different kind of storage system providing storage space for the archived data. In the next section we describe a solution that integrates iRODS with a tiered storage file system based on IBM Spectrum Scale and IBM Spectrum Archive.

TIERED STORAGE FILE SYSTEM

IBM Spectrum Scale™ [5] is a software-defined scalable parallel file system providing tiered storage capabilities. IBM Spectrum Archive Enterprise Edition [6] provides and manages the tape tier within an IBM Spectrum Scale file system. The IBM Spectrum Scale file system can be accessed via standardized protocols such as POSIX, NFS, SMB, HDFS and Object.

As shown in Figure 1, the combination of IBM Spectrum Scale with IBM Spectrum Archive provides a tiered storage file system with different storage media including Flash and SSD, disk and tape. While Flash and disk storage are managed by IBM Spectrum Scale directly, the tape storage is managed by IBM Spectrum Archive. IBM Spectrum Scale integrates a policy engine that allows to place the files on a storage tier upon file create and migrate the files to other storage tiers over the data lifecycle. Policies are defined and tested once and can then be configured to run automatically in the background. For example, a policy can place all new files on the disk storage tier of the IBM Spectrum Scale file system and if files have not been accessed for 30 days then migrate these files to tape storage.

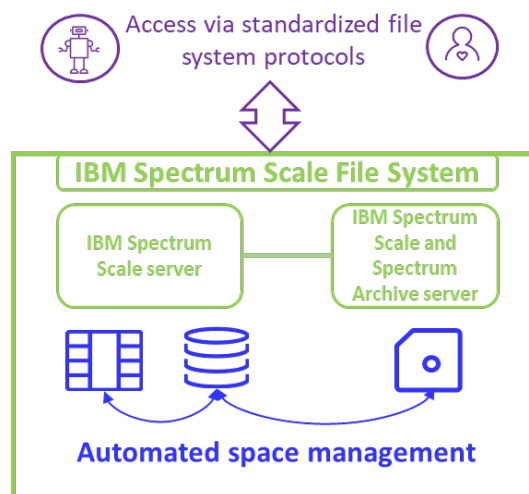


Figure 1: Combination of IBM Spectrum Scale with IBM Spectrum Archive tape tier

SOLUTION INTEGRATING IRODS WITH IBM SPECTRUM ARCHIVE

This solution integration iRODS and IBM Spectrum Archive is shown in Figure 2 and is comprised of three servers that are interconnected. One server represents the IBM Spectrum Scale cluster containing a tiered storage file system which is placed on disk and tape. The tape tier is managed by IBM Spectrum Archive. This file system is exported via NFS to the iRODS server.

The iRODS server hosts the iRODS Metadata Catalog (iCAT) database. The iCAT is a relational database that holds all the information about data, users, and zone that the iRODS servers need to facilitate the management and sharing of data.

The iRODS client can host an application that interacts with the iRODS server through the available API. In this example the iRODS client command line (iCommand) is used to archive, describe, search and retrieve data.

iRODS server, client and the NFS mounted tiered storage file system represent an iRODS zone.

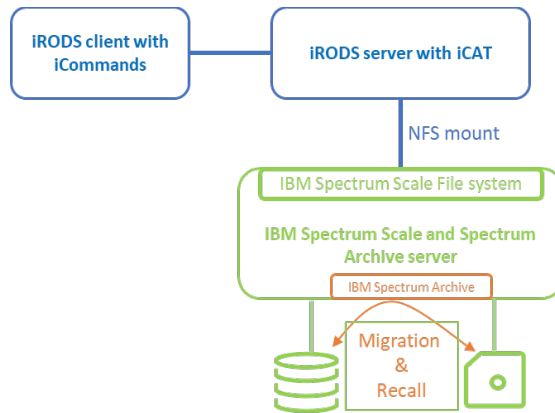


Figure 2: Solution architecture of iRODS with IBM Spectrum Archive

This solution can be configured to provide value adding functions, including:

- [Prevent transparent recalls](#) of files that are on tape and add files that are requested by the iRODS user to a queue and recall them in a tape optimized manner
- [Determine file migration state](#) from an iRODS user perspective
- [Set storage quota for users](#) that apply to all files regardless if stored on disk or on tape in the underlying
- [Extracting and ingesting file metadata](#) to the iRODS metadata catalog (iCAT) automatically after a file has been stored

Subsequently we briefly explain these functions. The actual implementation can be found at the GitHub repository [7].

Prevent transparent recall

To prevent transparent recalls, we can leverage a new iRODS rule along with a new custom microservice. The iRODS rule, runs on the iRODS server, intercepts an open request for a file using the system defined PEP rule `acPreprocForDataObjOpen` and invokes the new custom microservice along with the path and file name of the file to be opened. The new microservice determines if the file is migrated. If the file is not migrated, then the microservice returns “1” to the rule. Otherwise, if the file is migrated, then the microservice returns “0” to the rule and adds the path and filename queue. The queue can be a file list that resides on the IBM Spectrum Archive server. If the microservice returned “0” then the rule fails the file open request and informs the user that the file is still on tape.

Here is an example of a file open request for a migrated file:

```
$ iget -f file1
file /archive/home/mia/coll/file1 is still on tape, but queued to be staged.
```

To recall the files that have been added to the queue, a recall-program must be implemented that recalls these files using the tape optimized recall functions. This recall-program can be scheduled to run periodically on the IBM Spectrum Archive server, if the queue for the files to be recalled is a file list that is accessible by the Spectrum Archive server.

The time interval of the recall-program execution defines the maximum time the user must wait before he can access a file that was migrated to tape. To provide the user the capability to display the file status, we created another example which is explained next.

Display file status

To display the migration state of a file stored in an iRODS zone we created a new command for the iRODS user: `ifilestate`. This new command invokes a new iRODS rule that invokes a new microservice that checks the state of a file using the UNIX command: `stat`. Depending on the result of this check done by the new microservice the rule program returns the appropriate message to the user. Find below an example output of the new command:

```
$ ifilestate /archive/home/mia/coll/file1
Level 0: file /archive/home/mia/coll/file1 is MIGRATED

$ ifilestate /archive/home/mia/coll/file0
Level 0: file /archive/home/mia/coll/file0 is NOT migrated
```

Set quota for the entire file space

To set and enable quota for a given user using a given iRODS storage resource we did the following:

Enable quota by editing the file `/etc/core.re` and adding the following line:

```
acRescQuotaPolicy {msiSetRescQuotaPolicy("on"); }
```

Set quota limit of 2 GB for `user1` on the iRODS storage resource that represents the tiered storage file system provided by IBM Spectrum Scale. In this example we have one iRODS storage resource in the zone that is named “buffer”. Because we only have one storage resource the total quota limit is identical to the quota limit of the storage resource `buffer`:

```
$ iadmin suq user1 buffer 2147483648
$ iadmin suq user1 total 2147483648
```

To calculate the current storage consumption on a periodic basis we created a delayed iRODS rule that invokes the integrated microservice `msiQuota` and loaded this into the rule engine using the following command:

```
$ irule -F /etc/irods/quota.r -r irods_rule_engine_plugin-irods_rule_language-instance
```

Now if the user tries to store more than 2 GB on the storage resource he gets a quota exceeded error:

```
$ iput bigfile2
/archive/home/user1/coll/bigfile2, status = -110000 status = -110000
SYS_RESC_QUOTA_EXCEEDED
```

Extracting and ingesting metadata

The last project we implemented (essentially based on Daniel Moore NetCDF header extraction microservice [10]) extracts metadata from ingested files and add this into the iRODS catalog to make it available for subsequent searches.

For the implementation we again used a custom iRODS rules and microservice. We created a new iRODS rule that is invoked after a file has been stored in the iRODS zone, for example by using the `iput` command. This rule implements the integrated iRODS PEP `acPostProcForPut` and invokes a new microservice. The new microservice harvests the information from the file and return this to the iRODS rule which adds it to the file metadata.

To make it simpler in this paper, imagine the microservice determines the type of the file using the UNIX command: file and returns this as string to the iRODS rule. The iRODS rule adds the value of the file type string to the attribute Filetype to the file metadata. After ingesting files to iRODS using the iput command, the file will automatically obtain the file type as metadata as shown below:

```
$ iput document.pdf file1

$ imeta ls -d file1
AVUs defined for dataObj file1:
attribute: Filetype
value: PDF document
units:
```

It is also possible to search in iRODS for all files based on their type using the command: imeta:

```
$ imeta qu -d Filetype like %PDF%
collection: /archive/home/mia/coll
dataObj: file1
collection: /archive/home/mia/coll
dataObj: file2
```

As shown above the search found two files.

This is a simple example. There are many iRODS projects that leverage this mechanism to extract file header information from JPEG-files [8] or NETCDF-files [9] and many other file types.

REFERENCES

- [1] Disk and Tape TCO study by ESG:
<https://www.lto.org/wp-content/uploads/2018/08/ESG-Economic-Validation-Summary.pdf>
- [2] Blog article challenges and solutions with tiered storage file systems
<https://community.ibm.com/community/user/imwuc/blogs/nils-haustein1/2020/02/21/irods-with-ibm-spectrum-archive>
- [3] IBM Whitepaper “Integration of iRODS with IBM Spectrum Archive Enterprise Edition”
<http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102815>
- [4] iRODS: <https://irods.org/>
- [5] Spectrum Scale: https://en.wikipedia.org/wiki/IBM_Spectrum_Scale
- [6] Spectrum Archive: <https://developer.ibm.com/storage/products/ibm-spectrum-archive/>
- [7] Github repository for this project: <https://github.com/nhaustein/irods-tieredStorage-tape>
- [8] JPEG file: <https://en.wikipedia.org/wiki/JPEG>
- [9] NETCDF data format: <https://en.wikipedia.org/wiki/NetCDF>
- [10] Project for extracting NETCDF metadata by Daniel Moore: https://github.com/d-w-moore/extract_netcdf_header_msvc

DISCLAIMER

© Fondazione CMCC - Centro Euro-Mediterraneo sui Cambiamenti Climatici 2018

Visit www.cmcc.it for information on our activities and publications.

The Foundation Euro-Mediterranean Centre on Climate Change has its registered office and administration in Lecce and other units in Bologna, Venice, Capua, Sassari, Viterbo and Milan. The CMCC Foundation doesn't pursue profitable ends and aims to realize and manage the Centre, its promotion, and research coordination and different scientific and applied activities in the field of climate change study.

© IBM Corporation 2020

The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information provided, it is provided "as is" without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

The following terms are registered trademarks of International Business Machines Corporation in the United States and/or other countries: IBM Spectrum Scale, IBM Spectrum Archive

LINUX is a registered trademark of Linus Torvalds.

iRODS Copyright © 2005-2018, Regents of the University of California and the University of North Carolina at Chapel Hill. All rights reserved.

iRODS is released under a 3-clause BSD License.