iRODS S3 Resource Plugin: Glacier Support

Justin James
Renaissance Computing Institute (RENCI)
UNC Chapel Hill
jjames@renci.org

Terrell Russell
Renaissance Computing Institute (RENCI)
UNC Chapel Hill
unc@terrellrussell.com

ABSTRACT

The iRODS S3 Resource Plugin has been extended to honor the Glacier semantics of an S3 storage system including reacting appropriately to responses that indicate the data requested will be available later. This paper describes the implementation details and future work.

Keywords

iRODS, S3, glacier, storage

INTRODUCTION

The iRODS S3 Resource Plugin has been steadily improving over the years and growing new features, including cacheless behavior (removing the need to have it be configured as a child of a compound resource) [1], detached mode (which removes the need to redirect any S3 request to a particular iRODS server to service that request)[1], and direct streaming (allowing multipart put and get directly into the S3 layer) [2]. This year's update includes new support for Glacier semantics, similar to Amazon's own storage class behavior, but available in multiple other vendor's S3-compatible storage offerings.

OVERVIEW

The S3 storage service provided by Amazon offers multiple different storage classes [3]. The Glacier storage classes are the archive tiers for S3. While these classes and this interface and behavior are defined by the Amazon offering, other vendors have implemented the same. This new iRODS solution has been tested against Amazon and Fujifilm's Object Archive [4] product at this time.

The behavior is transparent (as compared to a regular iRODS transfer) except for a flag that is needed to define the storage class. This flag is part of the request and is required to signal the intent of the caller.

When downloading data, the behavior is asynchronous, except for storage classes that are deemed "instant retrieval". This is a vendor-specific definition and should be investigated and documented for any particular product and deployment. The iRODS S3 storage resource does not know what this definition may mean for any particular S3-compatible storage that may be configured for it to use.

GLACIER SUPPORT ON OBJECT READ

Adding Glacier support was relatively straightforward. Prior to reading an object from the iRODS namespace stored in an S3 resource, a `HeadObject` operation must be called to ascertain if the object is currently in archive within the S3 service. If the object is determined to be in the archive, then the object is requested and restoration will be scheduled. Otherwise, the object is returned immediately.

iRODS UGM 2022 July 5-8, 2022, Leuven, Belgium

Determining the status of the object is through the inspection of the `x-amz-storage-class` header:

- If it exists and is either `GLACIER` or `DEEP_ARCHIVE`, inspect the `x-amz-restore` header:
  - If `x-amz-restore` has `ongoing-request=true`, then a restore has already been scheduled. Return `REPLICA_IS_BEING_STAGED` error with message indicating the object is in process of being restored.
  - If `x-amz-restore` has `ongoing-request=false`, then the object has already been restored. Proceed as normal.
  - If `x-amz-restore` does not exist, the object is in archive. Call `RestoreObject` and return `REPLICA_IS_BEING_STAGED` with message indicating the object is being queued for restoration.
- If `x-amz-storage-class` header does not exist or is not `GLACIER` or `DEEP_ARCHIVE`, the object can be immediately retrieved. Proceed as normal.

RESTORING AN OBJECT FROM ARCHIVE

When it is determined that an object be restored from the archive, the `RestoreObject` operation is requested. To support this operation, two new resource context configuration settings have been introduced to the iRODS S3 resource plugin, `S3_RESTORATION_TIER` and `S3_RESTORATION_DAYS`. The use of these two configuration settings give the administrator full control over the behavior of the object restoration.

`S3_RESTORATION_TIER` - This is the value sent in the `<tier>` tag when `RestoreObject` is called. The values are not case sensitive. Valid values are 'Standard', 'Bulk', and 'Expedited'. The restoration tier, in combination with the storage class, defines the length of time needed to complete the restoration. The following are the restoration times for Amazon's S3 service:

	Glacier	**Deep Archive**
Expedited	1-5 minutes	Not Allowed
Standard (default)	3-5 hours	Within 12 hours
Bulk	5-12 hours	Within 48 hours

Note: `RestoreObject` is neither necessary nor allowed for objects stored in `Glacier_IR`.

`S3_RESTORATION_DATS` - The number of days the object will be restored. The default in the S3 plugin is 7. (According to Amazon, this is overridden if you have the bucket set up with lifecycle configuration.)

GLACIER SUPPORT ON WRITE OR COPY

For writing into the S3 resource, a single new resource context setting named `S3_STORAGE_CLASS` is provided. It is used to define the destination storage class for uploaded data objects and one of four valid values must be set (these are not case-sensitive):

- `STANDARD` - default
- `GLACIER`
- `DEEP_ARCHIVE`
- `GLACIER_IR` - Glacier Instant Retrieval

If defined, this setting is sent in the `x-amz-storage-class` header for `PutObject` and `CopyObject`.

This header may also have the following values which are either not relevant for Glacier support or have not yet been implemented:

- `STANDARD_IA` - Standard Infrequent Access
- `ONEZONE_IA` - One Zone Infrequent Access
- `INTELLIGENT_TIERING`
- `OUTPOST`

CHANGES TO LIBS3

The `libs3` library did not have support for Glacier and Deep Archive at the beginning of this work. The iRODS fork of the `libs3` library has been updated and the following three changes have been incorporated [5]:

- Implemented the `RestoreObject` API
- Added the ability to set `x-amz-storage-class` header on `PutObject` and `CopyObject`
- Added the ability to read `x-amz-storage-class` and `x-amz-restore` headers from the `HeadObject` header

We are planning to open a pull request to the upstream `libs3` library with these changes.

EXAMPLE GLACIER SETUP AND FILE RETRIEVAL

The following is an example of the configuration necessary to use the new Glacier behavior.

This configuration creates a resource that places files in Glacier, performs expedited restorations, and restores for 1 day.

```
$ iadmin mkresc s3resc s3 ‘hostname‘:/justinkylejames-irods1/amazons3resc \
"S3_DEFAULT_HOSTNAME=s3.amazonaws.com;S3_AUTH_FILE=/var/lib/irods/amazon.keypair;
S3_REGIONNAME=us-east-1;S3_PROTO=HTTP;HOST_MODE=cacheless_attached;
S3_STORAGE_CLASS=Glacier;S3_RESTORATION_TIER=Expedited;S3_RESTORATION_DAYS=1"
Creating resource:
Name:           "s3resc"
Type:           "s3"
Host:           "ce61bbc3beec"
Path:           "/justinkylejames-irods1/amazons3resc"
Context:        "S3_DEFAULT_HOSTNAME=s3.amazonaws.com;S3_AUTH_FILE=/var/lib/irods/amazon.keypair;
S3_REGIONNAME=us-east-1;S3_PROTO=HTTP;HOST_MODE=cacheless_attached;S3_STORAGE_CLASS=Glacier;
S3_RESTORATION_TIER=Expedited;S3_RESTORATION_DAYS=1"
```

To begin, create a local file (`test.txt`) and then put it into iRODS onto the newly created and configured S3 storage resource.

```
$ echo test123 > test.txt
$ iput -R s3resc test.txt
```

Next, try to get the object. The error code and error message lets the user know that the object is in the Glacier archive, has been queued for restoration within the S3 fabric, and to come back again and try to retrieve the file later.

```
$ iget test.txt -
remote addresses: 172.17.0.2 ERROR: getUtil: get error for - status = -721000 REPLICA_IS_BEING_STAGED
Level 0: [-]    /github/irods_resource_plugin_s3/s3/s3_transport/src/s3_transport.cpp:208:irods::error
irods::experimental::io::s3_transport::restore_s3_object(const std::string &,
libs3_types::bucket_context &, const unsigned int, const std::string &, const std::string &) :
status [REPLICA_IS_BEING_STAGED]  errno [] --
message [Object is in GLACIER and has been queued for restoration.  Try again later.]
```

Then, try to get the object a second time. The error code returned is the same as before, since the object is still not yet available, but the error message reflects the slightly different situation within the S3 fabric.

```
$ iget test.txt -
remote addresses: 172.17.0.2 ERROR: getUtil: get error for - status = -721000 REPLICA_IS_BEING_STAGED
Level 0: [-]    /github/irods_resource_plugin_s3/s3/s3_transport/src/s3_transport.cpp:133:irods::error
irods::experimental::io::s3_transport::handle_glacier_status(const std::string &,
libs3_types::bucket_context &, const unsigned int, const std::string &,
irods::experimental::io::s3_transport::object_s3_status, const std::string &) :
status [REPLICA_IS_BEING_STAGED] errno [] --
message [Object is in GLACIER and is currently being restored.  Try again later.]
```

Finally, wait a few minutes and attempt to retrieve the object again. The file is returned cleanly without any errors.

```
$ iget test.txt -
test123
```

STATUS AND FUTURE WORK

The iRODS S3 Resource Plugin has learned the Glacier semantics and has been partially released. Restoration from Glacier was added in 4.2.11.0 and included in 4.3.0.0.

The next release (4.3.0.1) will include support for setting the storage class on `PutObject` and `CopyObject` which will work with 'Deep Archive' for the put, get, and copy operations within the iRODS namespace.

After that, we expect that support for the intelligent tiering storage class should be trivial but this has not yet been implemented or tested.

Additionally, we could write a server-side iRODS rule to read metadata on an atomic put and select the storage class dynamically for object level control.

REFERENCES

[1] James, J., Russell, T., Coposky, J.: iRODS S3 Resource Plugin: Cacheless and Detached Mode. pp65-71. 2019 iRODS User Group Meeting. (2019)

[2] James, J., Draughn, K., Coposky, J., Russell, T. : S3:TNG - iRODS S3 Resource Plugin with Direct Streaming. pp15-24. 2020 iRODS User Group Meeting. (2020)

[3] Amazon S3 Glacier storage classes. `https://aws.amazon.com/s3/storage-classes/glacier/`

[4] Fujifilm Object Archive. `https://datastorage-na.fujifilm.com/object-archive/`

[5] James, J.: libs3 - Add glacier support. `https://github.com/irods/libs3/issues/16`