Summary of Software development for the Rucio Scientific Data Management system (as IRIS-HEP Fellows project)

Lev Pambuk,
Odesa National University of Technology
Mentors: Martin Barisits (CERN EP), Mario Lassnig (CERN EP)
Period: Jun – Sep, 2023
Outline

- Project & Objectives
  - Introductory work
  - Client developments
  - Operations developments
  - Core developments
  - Transfer developments

- Introduced:
  - Enhancements
  - Bug fixes
  - New features
  - Deletions
1. Correcting typos in docstrings for Python API of DIDClient

2. Removing link to deprecated Google Form in Rucio WebUI

3. Rearranging of imported statements in files and improving test
4. Improving error reporting during upload when reusing a deleted DID

Description

What the operators see:

Database exception.
Details: (raised as a result of Query-invoked autocommit; consider using a session'autocommit' block if this issue persists.)
ORA-20001: Primary key constraint DELETED_DIDS_PK violated
ORA-00051: at "ATLAS_RUCIO_CHECK_DID_UNIQUENESS", line 12
ORA-06003: error during execution of trigger "ATLAS_RUCIO_CHECK_DID_UNIQUENESS"
[SQL: INSERT INTO atlas_rucio.dids (scope, name, account, did_type, is_open, complete, availability, bytes, len) parameters: ('scope': 'XXX', 'name': 'XXX', 'account': 'XXX', 'did_type': 'F', 'is_open': 'ON', 'complete': 'NO')
(Background on this error at: https://sqlalchemy.org/34/bug)]

What the user sees:

2023-02-18 08:10:23,784 INFO Preparing upload for file XXX
2023-02-18 08:10:23,866 ERROR Database exception.
Details: An unknown Database Exception has occurred.

Motivation

The user is unable to deduce what is going wrong.

Kweizer commented on Aug 14

I put error handling of this violation in post method of dids.py.
So while uploading and reusing a deleted DID error handling will happen on the server side in more precise rest endpoint.
5. Webdav Protocol stat didn’t return data as specified

**Motivation**

The docstring for `res/protocols/webdav.py::Default::stat` states that a dict with two keys (filesize and adler32) are returned. This matches the implementations of `stat` in the SSH and xrootd protocols (although these protocols do not explicitly say that it is adler32 but any one of `GLOBALLY_SUPPORTED_CHECKSUMS`). At the moment, the implementation of webdav only returns the filesize.

**Modification**

Add an entry to the dictionary containing the checksum.
6. Rucio client lists replicas in wrong order
7. register-after-upload doesn't transition existing replicas to Available

Clients: Transition replicas to Available #6278 #6315

register-after-upload doesn't transition existing replicas to

rcarpa opened this issue on Jul 21 · 0 comments

rcarpa commented on Jul 21 · edited ·

When this option is set to `register upload`, this code is executed:

```
rucio/librucio/client/uploadclient.py
Line 209 in rucio
209 if register_after_upload;
```

but, `register_file` doesn't do anything if the replica exists:

```
rucio/librucio/client/uploadclient.py
Line 408 in rucio
408 if rse not in replica['status']["status"]: Even if it’s not in the available state. We should transition the replica to available if it’s not already in this state.
8. Allow to declare suspicious replicas by RSE and LFN

Allow to declare suspicious replicas by RSE and LFN #5906

Open nsmith- opened this issue on Oct 6, 2022 · 0 comments

nsmith- commented on Oct 6, 2022

Motivation

In #5392 declare_bad_file_replicas was modified to allow either a list of PFNs (string) or a list of replicas (‘scope’: ‘name’: ‘rse_id’: <rse_id> or “rse”: <rse_name>) as inputs. The declare_suspicious_file_replicas API does not have the same feature.

Modification

declare_suspicious_file_replicas should be modified to have the same feature.

Enhancement: Modified declare_suspicious_file_replicas #5906

Open Kwaizer wants to merge 1 commit into rucio:master from Kwaizer:patch-5906-declare_rse_lfn
9. Subscription: Adding check for subscription #6233

Description
Currently when you try to ask Rucio if a user has Subscriptions and they have none, a SubscriptionNotFound exception is intentionally raised.

This has been very confusing for myself and other users as we learn to use Subscriptions. It would be much clearer if this just indicated that there were no subscriptions for this user, without the scary looking traceback.

```
@bacode=\n(rucio-client) [bjwhite@GALACTICA:/tmp/rucio-client]$ rucio-admin subscription list --account root
Subscriptions not found.
Details: Subscription for account 'root' named 'None' not found
Rucio exited with an unexpected/unknown error, please provide the traceback below to the developers.
Traceback (most recent call last):
```

### Possible Solution

```
if len(subscriptions) == 0:
    print('There are no subscriptions for this account')
else:
    for sub in subscriptions:
        if sub['account'] == 'root' and sub['name'] == 'None':
            print(sub['id'])
        else:
            print(sub)
```

return SUCCESS

### Code Snippet

```python
lib/rucio/client/subscriptions/client.py
```
10. Core & Internals: rse_settings dictionary

**Motivation**

At the moment, the `rsemagers` takes most of its settings from a dictionary, often called `rse_settings`, eg in `rsemagers.py#create_protocol`. This dictionary is unnecessarily complex and hence obscure. It is also not easy to type-hint.

**Modification**

It would be minimally invasive to type-hint the dictionary, which is possible starting from Python 3.8 – but we currently still want to support py3.6.

I would suggest replacing it with a `NamedTuple` (added in py2.6 or py3.0, respectively) (with the benefit of being immutable, easy to type-hint, and more transparent), or possibly even a custom `dataclass`.

---

**Kwalzer commented on Aug 8**

As supporting of py3.6 was dropped, type-hinting has become possible. So I considered using it for editing `rse_settings` dict parameter of `rsemagers` functions.
To be frank, I think it's actually better to use `dict[str, Any]`. It creates a placeholder for future work using `typing.Dict` to unequivocally define the structure and expected content of dictionaries. We've discussed it multiple times, but haven't started working on it.
11. Adding size information to list_rules()
12. Improper use of NoResultFound

Description
We need to review all uses of NoResultFound because there's at least one that is misleading (it will never be raised):

```python
christidis commented on Aug 11 • edited

db/models/core/ddi/ddy.py

```12```

```{bash}
    with Stmt:
        models.DataIdentifierAssociation. "INDEX(CONTENTS CONTENTS_PM)". 'oracle'
    .filter_by(
        scope_id=scope,
        name=name
    )
    for tmp_dd in session.execute(stmt).yield_per(tmp_dd).scalars():
        yield {'scope': tmp_dd.child_scope, 'name': tmp_dd.child.name, 'type': tmp_dd.child}
        'byte': tmp_dd.bytes, 'editorId': tmp_dd.editorId, 'text': tmp_dd.text}
except NoResultFound:
    raise exception.DataIdentifierNotFound("Data identifier '%(scope)s%(name)s' not found")
```

Steps to reproduce

```python
$ echo 'import InternalScope'
$ python -c "from rucio.core.did import list_content; list([list_content(scope=InternalScope('foo'), name='bar'))
```
Thank you for your attention!