

Arvados - Bug #3147

[SDKs] Python clients should automatically retry failed API and Keep requests (including timeouts), in order to survive temporary outages like server restarts and network blips.

07/02/2014 09:41 AM - Peter Amstutz

Status:	Resolved	Start date:	08/22/2014
Priority:	Normal	Due date:	
Assigned To:	Brett Smith	% Done:	100%
Category:	SDKs	Estimated time:	0.00 hour
Target version:	2014-10-08 sprint		

Description

Desired behavior:

- Transactions that time out or produce 5xx errors should be reattempted after a delay
- Transactions that produce 4xx errors should not be reattempted

Background/example:

Keep services were restarted while doing an upload, which resulted in them being temporarily unavailable. Arv-put fails in this case (and crashes with an exception) instead of retrying for a bit.

Exception in thread Thread-72:

Traceback (most recent call last):

```
File "/usr/lib/python2.7/threading.py", line 552, in __bootstrap_inner
    self.run()
File "/usr/local/lib/python2.7/dist-packages/arvados/keep.py", line 213, in run
    body=self.args['data'])
File "/usr/local/lib/python2.7/dist-packages/httplib2/__init__.py", line 1593, in request
    (response, content) = self._request(conn, authority, uri, request_uri, method, body, headers,
    redirections, cachekey)
File "/usr/local/lib/python2.7/dist-packages/httplib2/__init__.py", line 1335, in _request
    (response, content) = self._conn_request(conn, request_uri, method, body, headers)
File "/usr/local/lib/python2.7/dist-packages/httplib2/__init__.py", line 1300, in _conn_request
    conn.connect()
File "/usr/local/lib/python2.7/dist-packages/httplib2/__init__.py", line 913, in connect
    raise socket.error, msg
error: [Errno 111] Connection refused
```

Traceback (most recent call last):

```
File "/usr/local/bin/arv-put", line 4, in <module>
    main()
File "/usr/local/lib/python2.7/dist-packages/arvados/commands/put.py", line 376, in main
    path, max_manifest_depth=args.max_manifest_depth)
File "/usr/local/lib/python2.7/dist-packages/arvados/commands/put.py", line 292, in write_direct
    ory_tree
    path, stream_name, max_manifest_depth)
File "/usr/local/lib/python2.7/dist-packages/arvados/collection.py", line 270, in write_director
    y_tree
    self.do_queued_work()
File "/usr/local/lib/python2.7/dist-packages/arvados/collection.py", line 197, in do_queued_work
    self._work_file()
File "/usr/local/lib/python2.7/dist-packages/arvados/collection.py", line 210, in _work_file
    self.write(buf)
File "/usr/local/lib/python2.7/dist-packages/arvados/collection.py", line 494, in write
    return super(ResumableCollectionWriter, self).write(data)
File "/usr/local/lib/python2.7/dist-packages/arvados/collection.py", line 281, in write
    self.flush_data()
File "/usr/local/lib/python2.7/dist-packages/arvados/commands/put.py", line 268, in flush_data
    super(ArvPutCollectionWriter, self).flush_data()
File "/usr/local/lib/python2.7/dist-packages/arvados/collection.py", line 286, in flush_data
    self._current_stream_locators += [Keep.put(data_buffer[0:self.KEEP_BLOCK_SIZE])]
File "/usr/local/lib/python2.7/dist-packages/arvados/keep.py", line 119, in put
```

```
return Keep.global_client_object().put(data, **kwargs)
File "/usr/local/lib/python2.7/dist-packages/arvados/keep.py", line 485, in put
    (data_hash, want_copies, have_copies))
arvados.errors.KeepWriteError: Write fail for 6afcd3c55f8c02043815464f33e4d52a: wanted 2 but wrote
1
```

Subtasks:

Task # 3735: Make crunch-job retry when it calls arv-put	Resolved
Task # 3802: arv-mount exposes retry support	Resolved
Task # 3658: PySDK Collection classes expose retry support	Resolved
Task # 3659: arv-put exposes retry support	Resolved
Task # 3657: Review 3147-pysdk-retries-wip	Resolved
Task # 3872: Review 3147-py-collection-retries-wip2	Resolved

Related issues:

Related to Arvados - Bug #3351: [SDK] arv-put hangs during remote upload	Resolved	07/28/2014
Related to Arvados - Bug #3419: [SDKs] Perl client library should retry faile...	Closed	09/23/2014
Related to Arvados - Story #3795: [Crunch/SDKs] Tasks need more retry support	Closed	09/03/2014
Related to Arvados - Bug #12684: [Python SDK] Retry on HTTP 5xx errors	New	

Associated revisions

Revision 728ee977 - 08/26/2014 11:32 AM - Brett Smith

Merge branch '3147-pysdk-retries'

Closes #3147.

Revision a4ab52b4 - 09/18/2014 07:56 PM - Brett Smith

Merge branch '3147-py-collection-retries-wip2'

Closes #3147.

Revision c23de360 - 09/18/2014 08:38 PM - Brett Smith

3147: Rebuild the Python SDK to satisfy FUSE version requirement.

When I wrote the FUSE driver's version dependency, I did my math wrong, and it's newer than any version of the SDK. This commit will cause a new SDK to be built with a new version number that satisfies the dependency. Refs #3147.

Revision 9d724d20 - 09/18/2014 09:24 PM - Brett Smith

3147: FUSE requires a built Python SDK.

Unlike our Ruby Gems, Python package versioning is sensitive to commits that actually affect the package, rather than the entire arvados repository. On the plus side, this makes it a little easier to reliably name a package version that meets our requirements. This commit does that. Refs #3147.

History

#1 - 07/02/2014 12:50 PM - Peter Amstutz

- Description updated

#2 - 07/02/2014 09:30 PM - Tom Clegg

- Subject changed from *arv-put should try harder when keep servers are temporarily unavailable.* to *Clients should automatically retry failed API and Keep requests, in order to survive temporary outages like server restarts and network blips.*

- Story points set to 3.0

#3 - 07/03/2014 02:25 PM - Tom Clegg

- Subject changed from *Clients should automatically retry failed API and Keep requests, in order to survive temporary outages like server restarts and network blips.* to *Clients should automatically retry failed API and Keep requests (including timeouts), in order to survive temporary outages like server restarts and network blips.*

#4 - 07/04/2014 03:04 PM - Tom Clegg

- Target version set to 2014-08-06 Sprint

#5 - 07/04/2014 03:43 PM - Tom Clegg

- Subject changed from *Clients should automatically retry failed API and Keep requests (including timeouts), in order to survive temporary outages like server restarts and network blips.* to *[SDKs] Clients should automatically retry failed API and Keep requests (including timeouts), in order to survive temporary outages like server restarts and network blips.*

- Description updated

- Category set to SDKs

#6 - 07/04/2014 09:30 PM - Peter Amstutz

In particular, when a job needs to upload output to Keep, it desirable that it keep retrying for at least as long as the job took to run up to that point. When there is an error uploading to Keep it should log enough information to diagnose what is wrong (possibly in a standard format that can be recognized by automated tools and raise a signal to operations). It should also keep refreshing the list of Keep services from API server in the event that servers have been removed/added.

#7 - 07/16/2014 04:38 PM - Peter Amstutz

- Assigned To set to Peter Amstutz

#8 - 07/16/2014 08:52 PM - Tom Clegg

- Target version deleted (2014-08-06 Sprint)

#9 - 07/28/2014 11:42 AM - Tom Clegg

- Target version set to 2014-08-27 Sprint

#10 - 07/29/2014 10:27 AM - Abram Connelly

When running the new arv-put from <https://arvados.org/attachments/download/332/arvados-python-client-0.1.1406400797.673a10d.tar.gz> (taken from the notes on bug [#3351](#)), I get:

```
WARNING:root:Retrying: PUT https://keep.qr1hi.arvadosapi.com:443/ aab838e794372e7da6eb59285f44ac7a
239808M / 293599M 81.7% WARNING:root:Request fail: PUT https://keep.qr1hi.arvadosapi.com:443/3cbd4e5e12803c392
1c3e941e0e0e22a => <class 'httplib.BadStatusLine'>: ''
WARNING:root:Retrying: PUT https://keep.qr1hi.arvadosapi.com:443/ 3cbd4e5e12803c3921c3e941e0e0e22a
Exception in thread Thread-3794:
Traceback (most recent call last):
  File "/usr/lib/python2.7/threading.py", line 810, in __bootstrap_inner
    self.run()
  File "/usr/local/lib/python2.7/dist-packages/arvados/keep.py", line 200, in run
    self.run_with_limiter(limiter)
  File "/usr/local/lib/python2.7/dist-packages/arvados/keep.py", line 221, in run_with_limiter
    body=self.args['data'])
  File "/usr/lib/python2.7/dist-packages/httplib2/__init__.py", line 1569, in request
    (response, content) = self._request(conn, authority, uri, request_uri, method, body, headers, redirections
, cachekey)
  File "/usr/lib/python2.7/dist-packages/httplib2/__init__.py", line 1316, in _request
    (response, content) = self._conn_request(conn, request_uri, method, body, headers)
  File "/usr/lib/python2.7/dist-packages/httplib2/__init__.py", line 1251, in _conn_request
    conn.connect()
  File "/usr/lib/python2.7/dist-packages/httplib2/__init__.py", line 1059, in connect
    raise socket.error, msg
error: [Errno 111] Connection refused

Traceback (most recent call last):
  File "/usr/local/bin/arv-put", line 4, in <module>
    main()
  File "/usr/local/lib/python2.7/dist-packages/arvados/commands/put.py", line 434, in main
    path, max_manifest_depth=args.max_manifest_depth)
  File "/usr/local/lib/python2.7/dist-packages/arvados/commands/put.py", line 302, in write_directory_tree
    path, stream_name, max_manifest_depth)
  File "/usr/local/lib/python2.7/dist-packages/arvados/collection.py", line 270, in write_directory_tree
    self.do_queued_work()
  File "/usr/local/lib/python2.7/dist-packages/arvados/collection.py", line 197, in do_queued_work
    self._work_file()
  File "/usr/local/lib/python2.7/dist-packages/arvados/collection.py", line 210, in _work_file
    self.write(buf)
  File "/usr/local/lib/python2.7/dist-packages/arvados/collection.py", line 494, in write
    return super(ResumableCollectionWriter, self).write(data)
```

```
File "/usr/local/lib/python2.7/dist-packages/arvados/collection.py", line 281, in write
    self.flush_data()
File "/usr/local/lib/python2.7/dist-packages/arvados/commands/put.py", line 278, in flush_data
    super(ArvPutCollectionWriter, self).flush_data()
File "/usr/local/lib/python2.7/dist-packages/arvados/collection.py", line 286, in flush_data
    self._current_stream_locators += [Keep.put(data_buffer[0:self.KEEP_BLOCK_SIZE])]
File "/usr/local/lib/python2.7/dist-packages/arvados/keep.py", line 120, in put
    return Keep.global_client_object().put(data, **kwargs)
File "/usr/local/lib/python2.7/dist-packages/arvados/keep.py", line 516, in put
    (data_hash, want_copies, have_copies))
arvados.errors.KeepWriteError: Write fail for 3cbd4e5e12803c3921c3e941e0e0e22a: wanted 2 but wrote 0
```

#11 - 07/29/2014 10:32 AM - Abram Connelly

Restarting arv-put (again, as installed from <https://arvados.org/attachments/download/332/arvados-python-client-0.1.1406400797.673a10d.tar.gz>), the upload does not resume from the last block uploaded but restarts the upload.

#12 - 07/30/2014 11:33 AM - Tom Clegg

- Subject changed from [SDKs] Clients should automatically retry failed API and Keep requests (including timeouts), in order to survive temporary outages like server restarts and network blips. to [SDKs] Python clients should automatically retry failed API and Keep requests (including timeouts), in order to survive temporary outages like server restarts and network blips.

- Story points changed from 3.0 to 2.0

#13 - 07/30/2014 01:47 PM - Tom Clegg

- Description updated

#14 - 07/30/2014 01:47 PM - Tom Clegg

- Assigned To deleted (Peter Amstutz)

#15 - 08/06/2014 03:16 PM - Brett Smith

- Assigned To set to Brett Smith

#16 - 08/13/2014 02:42 PM - Brett Smith

- Status changed from New to In Progress

#17 - 08/21/2014 05:33 PM - Brett Smith

This story has an interesting interaction with Keep GET requests. The SDK's current algorithm for that is: Make a list of request URLs, first from known Keep services, then from Keep servers listed in location hints. Try each one in sequence until one succeeds, and return that result. If none succeed, raise a `NotFoundError`.

Since we're talking to different servers, I wanted clarification about how different kinds of errors affected the retry loop. Tom and I discussed the following implementation, in vague code:

```
responses = {}
for try_count in retries:
    requests = []
    for server in server_list():
        if not permanent_failure(responses.get(server)):
            requests.append(server)
    if not requests:
        break
    for server in requests:
        back_off_for(server) # Avoid request flooding
        response = get(server + block_path)
        if is_success(response):
            return response
        else:
            responses[server] = response
fail_count = len(responses)
permfail_count = len(key for key in responses
                    if permanent_failure(responses[key]))
if (permfail_count / fail_count) > .75: # Or some other threshold
    raise NotFoundError
else:
    raise KeepReadError
```

#18 - 08/22/2014 04:35 PM - Brett Smith

3147-pysdk-retries-wip is up for review. This adds retry support to KeepClient, which is the lowest-level part of the job.

A couple of notes about the design:

- This adds the mock library to test dependencies. But the retry tests run instantly, so the additional complication seems worth it.
- It may help to read `retry.py` before `keep.py`. It implements loop controller classes that do a lot of the heavy lifting of retry logic.
- We consistently use `num_retries` as the name of our parameter. The `apiclient` library that we use for the API SDK already supports this parameter in the `execute` method, so I wanted the consistency. (This is especially why it's `num_retries` rather than `num_tries`).
- I had to count 403 as a "Not Found" response for GET requests in order to for Keep permission tests to continue passing. But perhaps the tests should be changed to expect `KeepReadError`, since 403 doesn't mean "not found?"

Tasks that remain to be done:

- Add retry flags to the higher-level parts of the SDK, namely `CollectionReader` and `CollectionWriter`.
- Expose retry support in command-line tools. `arv-put` is the big target for this sprint; other tools can come later.

#19 - 08/25/2014 05:48 PM - Tim Pierce

Review at [dca92e4](#)

The code review advice was really helpful here. Thanks for the summary. The testing here is outstanding.

Two questions, one minor, one possibly significant:

- `sdk/python/arvados/keep.py`
 - The `_logger.info()` call at L246 performs integer division on `len(content)/(1024*1024)`, so if content is less than 1 MiB this will return 0. It's a minor thing but will look weird and inconsistent in the log output.
 - `KeepClient.get():L607`: as I understand the saved results, the second element in this tuple should be the number of servers that we queried. I don't understand why `len(services_to_try)` is the correct number here. It looks to me as though, if `services_to_try` has five servers, and the first one returns a blob, this code will save the tuple (blob, 5) when we only actually asked one of them. Is this just semantic confusion or is there a functional issue here?

#20 - 08/25/2014 09:09 PM - Brett Smith

Tim Pierce wrote:

Two questions, one minor, one possibly significant:

- `sdk/python/arvados/keep.py`
 - The `_logger.info()` call at L246 performs integer division on `len(content)/(1024*1024)`, so if content is less than 1 MiB this will return 0. It's a minor thing but will look weird and inconsistent in the log output.

Yup. Coerced this to a float with formatting to help clarify. Thanks for catching that.

- `KeepClient.get():L607`: as I understand the saved results, the second element in this tuple should be the number of servers that we queried. I don't understand why `len(services_to_try)` is the correct number here. It looks to me as though, if `services_to_try` has five servers, and the first one returns a blob, this code will save the tuple (blob, 5) when we only actually asked one of them. Is this just semantic confusion or is there a functional issue here?

You understand correctly, based on the comment in `_check_loop_result`. However, I think this is a comment bug rather than a code bug. What `_check_loop_result` really cares about is, "If this loop couldn't send the request to any servers, stop the loop, because future iterations will do the same (and we'll end up sleeping for no reason)." It doesn't distinguish between "servers eligible to receive the request" and "servers that actually received the request." I've tried to clarify the comment to make this less confusing for future readers. Ready for another look at [a46654d](#). Thanks.

#21 - 08/26/2014 09:36 AM - Tim Pierce

LGTM, thanks. Good to merge. Thanks for adding the 403 as an open question -- this doesn't seem urgent enough to spend a lot of time on right now, but we should be mindful of it.

#22 - 08/26/2014 11:35 AM - Brett Smith

- Status changed from *In Progress* to *Resolved*

- % Done changed from 17 to 100

Applied in changeset `arvados|commit:728ee9775de422840b81827c71c7b515708620a3`.

#23 - 08/26/2014 03:53 PM - Brett Smith

- Status changed from *Resolved* to *In Progress*

#24 - 08/27/2014 03:16 PM - Brett Smith

- Target version changed from *2014-08-27 Sprint* to *2014-09-17 sprint*

- Story points changed from 2.0 to 1.0

Carried forward to following sprint with 1 story point.

#25 - 09/16/2014 08:32 AM - Brett Smith

- Status changed from In Progress to New

3147-py-collection-retries-wip is up for review to implement the remaining requirements in this story. A couple of pointers:

- You might want to review the `retry_method` decorator in `retry.py`, and the illustrative tests in `test_retry.py`, before anything else. It makes sense on its own and gets used throughout the rest of the SDK.
- The Keep retry work was done in a previous sprint/branch, so you won't see that (except touch-ups for consistency).
- I did remove some stale code while I was in here. Some error handling in `arv-get` was redundant with error handling in `CollectionReader`, so it's gone. Some logger setup code in `arv-mount` was accommodating older versions of the Python SDK; now that I made the FUSE driver specify a dependency on a very recent SDK, that's no longer necessary.

#26 - 09/16/2014 08:32 AM - Brett Smith

- Status changed from New to In Progress

#27 - 09/18/2014 06:08 PM - Brett Smith

- Target version changed from 2014-09-17 sprint to 2014-10-08 sprint

#28 - 09/18/2014 06:43 PM - Tim Pierce

Review at [c19503ef3](#):

This is probably a dumb question, but can the api client object be instantiated with a default `num_retries` setting, like

```
api = arvados.api('v1', num_retries=args.num_retries)
```

That seems like it would be a lot cleaner than having to supply a `num_retries` argument to every `.execute()` call, e.g. `parent_project_uuid = api.users().current().execute(num_retries=args.retries)['uuid']`

Is the problem here that the Google API client library doesn't give us a good way to inject `num_retries` into methods defined via the discovery document?

`sdk/python/arvados/collection.py`

- Having `num_retries` be a publicly accessible member could lead to some confusion if the user thinks they can change `num_retries` after a keep client has been instantiated, e.g.

```
c = arvados.CollectionReader(manifest, num_retries=3)
# user reads some data from the collection here
c.num_retries = 5      # I changed my mind
for f in c.all_files:  # but the Keep client is still using num_retries=3
    ...
```

I'd recommend making `num_retries` a private field, and (if we want to expose it) supplying a setter that either sets `_keep_client=None` or changes its `num_retries` field to match.

`sdk/cli/bin/arv-get`

- It's a small optimization, but looks like it makes sense to move lines 137-138 (test if local file already exists) up to the top of `if not args.n -- you can avoid fetching the manifest at all from a possibly slow Keep server if you know you have to abort anyway.`

#29 - 09/18/2014 07:38 PM - Brett Smith

Tim Pierce wrote:

Review at [c19503ef3](#):

This is probably a dumb question, but can the api client object be instantiated with a default `num_retries` setting, like

It's a great question, but unfortunately the answer seems to be no. API methods return `HttpRequest` objects, and there's nothing about `num_retries` in its constructor. grepping the whole `apiclient` source code doesn't turn up anything promising, either.

Is the problem here that the Google API client library doesn't give us a good way to inject `num_retries` into methods defined via the discovery document?

No. `apiclient`'s `num_retries` support is purely a nicety of the client; it neither needs nor wants anything from the server. That's why it's an argument to `HttpRequest.execute()` rather than the actual API methods like `collections().list()`.

`sdk/python/arvados/collection.py`

- Having `num_retries` be a publicly accessible member could lead to some confusion if the user thinks they can change `num_retries` after a `keep` client has been instantiated, e.g.
[...]
I'd recommend making `num_retries` a private field, and (if we want to expose it) supplying a setter that either sets `_keep_client=None` or changes its `num_retries` field to match.

Per our discussion in the office, there's nothing in Python that's really as private as desired, and there are all kinds of ways to get mismatched values between different client objects. Added docstrings to the `Collection` objects to spell this out, since that's where it's most likely to come up for users.

`sdk/cli/bin/arv-get`

- It's a small optimization, but looks like it makes sense to move lines 137-138 (test if local file already exists) up to the top of `if not args.n --` you can avoid fetching the manifest at all from a possibly slow `Keep` server if you know you have to abort anyway.

This is how the code worked before I got here. I changed it to avoid a race condition where the file didn't exist at check time, but someone else wrote it while we were downloading the manifest. Your comment spurred me to fix it the right way, so now we open the file immediately and fail if it already exists. The error reporting got a little nicer as a consequence, too.

Now at [21cdefd](#). Thanks.

#30 - 09/18/2014 07:42 PM - Tim Pierce

Thanks for the comments and nice catch on the `arv-get` race. LGTM.

#31 - 09/18/2014 08:00 PM - Brett Smith

- *Status changed from In Progress to Resolved*

Applied in changeset `arvados|commit:a4ab52b4305023d8c679cc7c3b09361050d167b6`.

#32 - 11/28/2017 03:40 AM - Tom Morris

- *Related to Bug #12684: [Python SDK] Retry on HTTP 5xx errors added*