

## Arvados - Feature #7751

**[Crunch] [SDKs] [FUSE] Convenient way to write job output to Keep via writable arv-mount, as an alternative to staging output on scratch and then copying when finished.**

11/11/2015 03:19 PM - Tom Clegg

<b>Status:</b>	Resolved	<b>Start date:</b>	11/16/2015
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assigned To:</b>	Tom Clegg	<b>% Done:</b>	100%
<b>Category:</b>	SDKs	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	2015-12-16 sprint		
<b>Description</b>			
It is already <i>possible</i> for a crunch program to do this: start arv-mount in writable mode, write files into a new directory, and use the resulting PDH as the task output.			
This story makes it convenient to do this, i.e., the crunch script itself shouldn't need to do anything more complicated than this:			
<pre>outputdir = arvados.crunch.task_output_dir()  with open(os.path.join(outputdir.path, 'foo'), 'w') as f:     f.write('foo')  arvados.current_task().set_output(outputdir) # or perhaps just: outputdir.save()</pre>			
Possible implementation approach:			
<ul style="list-style-type: none"><li>• crunch-job sets up a writable fuse mount for every job task (but if the job doesn't do anything with it, nothing gets written; and it does not include any read or write access to existing collections beyond the by-PDH access already needed by jobs)</li><li>• add SDK functions that figure out (by looking at environment vars, etc.) where the output directory is supposed to go; push arv-mount's magic buttons<sup>1</sup> to get the PDH of the finished collection; and set the task output to that PDH.</li></ul>			
<sup>1</sup> Read JSON from {dir}/.arvados#collection			
<b>Subtasks:</b>			
Task # 7784: Review 7751-mount-tmp			<b>Resolved</b>
Task # 7792: CLI argument to add writable tmp collection to magic dir			<b>Resolved</b>
Task # 7827: Tests			<b>Resolved</b>
Task # 7865: Make .arvados#collection work reliably in jenkins			<b>Resolved</b>
Task # 7793: Python SDK helpers			<b>Resolved</b>
Task # 7872: Review 7751-crunch-fuse-output			<b>Resolved</b>
Task # 7873: Try new crunch-job on staging			<b>Resolved</b>
<b>Related issues:</b>			
Blocks Arvados - Feature #7847: [SDKs] Update run-command to use arv-mount --...			<b>Rejected</b>

### Associated revisions

#### Revision 346a5589 - 11/25/2015 03:45 PM - Tom Clegg

Merge branch '7751-mount-tmp' refs #7751

#### Revision 8b65fac8 - 11/25/2015 06:06 PM - Tom Clegg

7751: Clean up KEEP\_LOCAL\_STORE env var after tests.

refs #7751

#### Revision 5590c9ac - 12/03/2015 05:03 PM - Tom Clegg

Merge branch '7751-crunch-fuse-output' closes #7751

### History

#1 - 11/11/2015 03:55 PM - Tom Clegg

- Description updated

## #2 - 11/11/2015 03:55 PM - Tom Clegg

- Category set to SDKs

- Target version changed from Arvados Future Sprints to 2015-12-02 sprint

## #3 - 11/12/2015 03:34 PM - Brett Smith

- Story points set to 1.0

## #4 - 11/12/2015 03:40 PM - Brett Smith

- Assigned To set to Tom Clegg

## #5 - 11/12/2015 03:42 PM - Tom Clegg

- Story points changed from 1.0 to 2.0

## #6 - 11/20/2015 08:33 PM - Tom Clegg

- Status changed from New to In Progress

## #7 - 11/20/2015 11:46 PM - Tom Clegg

7751-mount-tmp @ [2732213](#) includes:

- "scratch space" mount mode: data blocks get written to Keep, but a collection never gets saved (at least not by arv-mount -- you can get a manifest by reading the magic .arvados#collection file before unmounting, and save that to a collection).
- Rearrange arv-mount args a bit so caller can specify a custom layout: --mount-tmp foo --mount-by-pdh by\_id /tmp/mnt → /tmp/mnt/foo is scratch space, /tmp/mnt/by\_id/ is magic by-id dir.
- Move big pile of code out of bin/arv-mount into the arvados\_fuse module dir
  - → argument-parsing and nearly all of the actual FUSE setup code is now used in the new integration tests (was untested before).
  - → now possible to test argument handling without the overhead of subscribing to websockets and bringing up a fuse mount
- Small bugfix for [#7654](#) where we accidentally monkey-patched a ws4py closed() method with a bool, causing a stack trace during websocket shutdown.

## #8 - 11/23/2015 08:32 PM - Peter Amstutz

Test failures:

```
=====
ERROR: runTest (tests.test_mount.MagicDirApiError)
-----
Traceback (most recent call last):
  File "/home/peter/work/arvados/services/fuse/tests/test_mount.py", line 1034, in runTest
    llfuse.listdir(os.path.join(self.mounttmp, self.testcollection))
  File "llfuse/fuse_api.pxi", line 43, in llfuse.capi.listdir (src/llfuse/capi_linux.c:22621)
OSError: [Errno 2] No such file or directory: '/tmp/tmpSbUVZB/97d180c4f916faf61fb3d64aa2263961+52'

=====
FAIL: test_tmp_snapshots (tests.test_tmp_collection.TmpCollectionTest)
-----
Traceback (most recent call last):
  File "/home/peter/work/arvados/services/fuse/tests/integration_test.py", line 66, in wrapper
    return func(self, *args, **kwargs)
  File "/home/peter/work/arvados/services/fuse/tests/test_tmp_collection.py", line 101, in test_tmp_snapshots
    self.pool_test(os.path.join(self.mnt, 'zzz'))
  File "/home/peter/work/arvados/services/fuse/tests/integration_test.py", line 34, in pool_test
    (modName, clsName, '_' + funcName, args, kwargs))
  File "/usr/lib/python2.7/multiprocessing/pool.py", line 244, in apply
    return self.apply_async(func, args, kwds).get()
  File "/usr/lib/python2.7/multiprocessing/pool.py", line 558, in get
    raise self._value
AssertionError: Regexp didn't match: '^\\. 37b51d194a7513e45b56f6524f2d51f2\\+3(\\+\\S+)? 0:3:bar\\n$' not found in u'. 37b51d194a7513e45b56f6524f2d51f2+3+A90de89bcf2ee65cbe320f2eac348858e7725be20e5665e6c3 acbd18db4cc2f85cedef654fccc4a4d8+3+A77c716f7ea2d409fdb116f8b4c52afb0013fcff@5665e6c3 0:3:bar 3:3:foo\n'
```

tests.test\_mount.MagicDirApiError is failing consistently. test\_tmp\_snapshot seems to be a race condition that only fails sometimes.

This came up as an exception but the test is still marked as success (?):

```
test_two_tmp (tests.test_tmp_collection.TmpCollectionTest) ... [keep1] 2015/11/23 15:23:55 [[:1]:58834] PUT a
cbd18db4cc2f85cedef654fccc4a4d8 0.000169s 200 86 "OK"
[keep0] 2015/11/23 15:23:55 [[:1]:42365] PUT acbd18db4cc2f85cedef654fccc4a4d8 0.000154s 200 86 "OK"
```

```

[keep1] 2015/11/23 15:23:55 [[::]:58836] PUT 37b51d194a7513e45b56f6524f2d51f2 0.000424s 200 86 "OK"
[keep0] 2015/11/23 15:23:55 [[::]:42367] PUT 37b51d194a7513e45b56f6524f2d51f2 0.002501s 200 86 "OK"
2015-11-23 15:23:56 arvados.arvados_fuse[16092] ERROR: Unhandled exception during FUSE operation
Traceback (most recent call last):
  File "/home/peter/work/arvados/services/fuse/arvados_fuse/__init__.py", line 277, in catch_exceptions_wrappe
r
    return orig_func(self, *args, **kwargs)
  File "/home/peter/work/arvados/services/fuse/arvados_fuse/__init__.py", line 472, in forget
    ent = self.inodes[inode]
  File "/home/peter/work/arvados/services/fuse/arvados_fuse/__init__.py", line 215, in __getitem__
    return self._entries[item]
KeyError: 1L
Exception in thread Thread-31:
Traceback (most recent call last):
  File "/usr/lib/python2.7/threading.py", line 810, in __bootstrap_inner
    self.run()
  File "/usr/lib/python2.7/threading.py", line 763, in run
    self.__target(*self.__args, **self.__kwargs)
  File "/home/peter/work/arvados/services/fuse/arvados_fuse/command.py", line 118, in <lambda>
    t = threading.Thread(None, lambda: llfuse.main())
  File "llfuse/fuse_api.pxi", line 317, in llfuse.capi.main (src/llfuse/capi_linux.c:25028)
  File "llfuse/handlers.pxi", line 56, in llfuse.capi.fuse_forget (src/llfuse/capi_linux.c:2730)
  File "llfuse/handlers.pxi", line 57, in llfuse.capi.fuse_forget (src/llfuse/capi_linux.c:2682)
  File "/home/peter/work/arvados/services/fuse/arvados_fuse/__init__.py", line 290, in catch_exceptions_wrappe
r
    raise llfuse.FUSEError(errno.EIO)
FUSEError: Input/output error

```

arvados\_fuse.command.Mount.Run should be lowercase run according to PEP8 (<https://www.python.org/dev/peps/pep-0008/#function-names>)

#### #9 - 11/23/2015 08:54 PM - Peter Amstutz

Turns out tests.test\_mount.MagicDirApiError is failing is master too, so that may be unrelated.

#### #10 - 11/23/2015 09:22 PM - Peter Amstutz

Peter Amstutz wrote:

Turns out tests.test\_mount.MagicDirApiError is failing is master too, so that may be unrelated.

My suspicion is that around v4.0 the Linux VFS may have changed the way directories listings are cached, so this test which checks if something exists (OSError on induced API fail) then checks again (API success) is failing the 2nd time because the 2nd request is being served from the Linux VFS cache instead of asking arv-mount again.

I'll file a separate issue.

#### #11 - 11/23/2015 09:43 PM - Peter Amstutz

Task [#7793](#) (Python SDK helpers) should include updating run-command.

#### #12 - 11/23/2015 09:45 PM - Peter Amstutz

bin/arv-mount is missing import arvados\_fuse.command

#### #13 - 11/23/2015 09:48 PM - Peter Amstutz

```

$ arv-mount --foreground --mount-tmp=foo ~/keep
Traceback (most recent call last):
  File "/home/peter/work/scripts/venv/bin/arv-mount", line 4, in <module>
    __import__('pkg_resources').run_script('arvados-fuse==0.1.20151122102613', 'arv-mount')
  File "/home/peter/work/scripts/venv/local/lib/python2.7/site-packages/pkg_resources/__init__.py", line 735,
in run_script
    self.require(requires)[0].run_script(script_name, ns)
  File "/home/peter/work/scripts/venv/local/lib/python2.7/site-packages/pkg_resources/__init__.py", line 1652,
in run_script
    exec(code, namespace, namespace)
  File "/home/peter/work/scripts/venv/lib/python2.7/site-packages/arvados_fuse-0.1.20151122102613-py2.7.egg/EG
G-INFO/scripts/arv-mount", line 8, in <module>
    arvados_fuse.command.Mount(args).Run()
  File "/home/peter/work/scripts/venv/local/lib/python2.7/site-packages/arvados_fuse-0.1.20151122102613-py2.7.
egg/arvados_fuse/command.py", line 130, in Run
    self._run_standalone(self.args)
TypeError: _run_standalone() takes exactly 1 argument (2 given)

```

#### #14 - 11/23/2015 10:27 PM - Tom Clegg

Peter Amstutz wrote:

tests.test\_mount.MagicDirApiError is failing consistently.

I haven't seen this one yet (but I've only tried ~5 times so far)

test\_tmp\_snapshot seems to be a race condition that only fails sometimes.

I see this one if I run it enough times.

It seems like unlink() is special here: this test (and several others) are deliberately testing race conditions and so far I haven't seen any other failures.

This came up as an exception but the test is still marked as success (?):

I see this one too, occasionally, and indeed it doesn't fail the test. Another shutdown race, I suppose. But so far I've only seen it in test\_two\_tmp, which also has an unlink() -- coincidence?

avados\_fuse.command.Mount.Run should be lowercase run according to PEP8 (<https://www.python.org/dev/peps/pep-0008/#function-names>)

Fixed @ [796c33a](#)

#### #15 - 11/23/2015 10:45 PM - Tom Clegg

Peter Amstutz wrote:

Task [#7793](#) (Python SDK helpers) should include updating run-command.

I'll probably just do the Python SDK helpers in [#7793](#). I've added [#7847](#) to make use of those in run-command (it obviously stands to benefit from this) but IIRC we deliberately left run-command out of this because the first jobs we want to try this on don't use run-command anyway. Also, it would be reassuring to see real-world results from some deliberate experiments before we start making all run-command jobs rely on it...

Total arv-mount breakage should be fixed as of [f6bd4a2](#) (whoops)

#### #16 - 11/24/2015 04:05 PM - Tom Clegg

The "KeyError: 1L" shutdown race is fixed in [4574dbb](#). Here's what happened:

- When "fusermount -u" is called from userspace, llfuse makes a list of inodes to forget, and makes multiple calls to operations.forget(). Each call to forget() has llfuse.lock, but the lock is released between calls.
- If operations.destroy() is called from arv-mount (or the test suite), it clears inodes. Therefore, subsequent calls to forget() -- e.g., arising from an external unmount that was already in progress -- will crash.

Fix:

- Acquire llfuse.lock in destroy() so it doesn't race forget().
- forget() is a no-op if shutdown has already cleared inodes.

#### #17 - 11/24/2015 04:13 PM - Peter Amstutz

I'm not keen on monkey-patching the collection object to intercept save() and save\_new(). The following approach would be more in line with the existing cache invalidation strategy used in the fuse driver:

1. Subclass Collection and override save() and save\_new() to "pass" (no-op)
2. TmpCollectionDirectory overrides on\_event() and calls self.inodes.invalidate\_inode() and self.collection\_record\_file.invalidate() then calls super()
3. Subclass ObjectFile and override read() and size() to regenerate self.contents from the collection when self.stale is True

The idea is to aggressively invalidate ".avados#collection" and generate an up-to-date version on demand rather than serving a cached copy.

Unfortunately, invalidate\_inode in llfuse is an asynchronous operation. This is because (in the kernel VFS) the invalidate operation acquires a lock on the inode; if you call invalidate\_inode synchronously on the current inode, it will result in a deadlock.

#### #18 - 11/24/2015 06:47 PM - Tom Clegg

Peter Amstutz wrote:

1. Subclass Collection and override save() and save\_new() to "pass" (no-op)
3. Subclass ObjectFile and override read() and size() to regenerate self.contents from the collection when self.stale is True

Refactored. [4c0d302](#)

2. TmpCollectionDirectory overrides on\_event() and calls self.inodes.invalidate\_inode() and self.collection\_record\_file.invalidate() then calls super()

Would CollectionDirectory.update() do anything for us here? I skipped CollectionDirectory and subclassed CollectionDirectoryBase instead because CollectionDirectory seems to be concerned with merging (which we don't want) and has its own .arvados#collection implementation that focuses on getting a current API server record (which we also don't want).

Calling invalidate\_inode() from the update() callback instead of unlink()→flush()→lock\_released→save() doesn't seem to affect the race bug at all. But invalidating the entry (not just the inode) does: I've updated the test to do 10 iterations of adding and removing files, which nearly always fails before that change, and hasn't failed yet after that change. [c333c47](#)

#### #19 - 11/24/2015 10:38 PM - Tom Clegg

Tom Clegg wrote:

Calling invalidate\_inode() from the update() callback instead of unlink()→flush()→lock\_released→save() doesn't seem to affect the race bug at all. But invalidating the entry (not just the inode) does: I've updated the test to do 10 iterations of adding and removing files, which nearly always fails before that change, and hasn't failed yet after that change. [c333c47](#)

Nope, that didn't work either. With 400 iterations that fails too. But setting llfuse.EntryAttributes().entry\_timeout = 0 survives 1400 iterations.

<http://pythonhosted.org/llfuse/data.html>

[9d80d31](#)

#### #20 - 11/25/2015 02:37 PM - Peter Amstutz

For the record, here's a trace of a failing test:

```
[...]
2015-11-25 09:26:49 arvados.arvados_fuse[21735] DEBUG: arv-mount unlink: 2 'foo'
2015-11-25 09:26:49 arvados.arvados_fuse[21735] DEBUG: collection notify del <arvados_fuse.fusedir.UnsaveableC
ollection object at 0x7f3006218150> foo <arvados.arvfile.ArvadosFile object at 0x7f3006334bd0>
2015-11-25 09:26:49 arvados.arvados_fuse[21735] DEBUG: del_entry on inode 3 with refcount 1
2015-11-25 09:26:49 arvados.arvados_fuse[21735] DEBUG: <arvados_fuse.fusedir.TmpCollectionDirectory object at
0x7f3006218190> invalidated collection record
2015-11-25 09:26:49 arvados.arvados_fuse[21735] DEBUG: arv-mount lookup: parent_inode 2 name '.arvados#collect
ion' inode 4
2015-11-25 09:26:49 arvados.arvados_fuse[21735] DEBUG: arv-mount forget: inode 3 nlookup 1 ref_count 1
2015-11-25 09:26:49 arvados.arvados_fuse[21735] DEBUG: arv-mount read 6 0 4096
2015-11-25 09:26:49 arvados.arvados_fuse[21735] DEBUG: arv-mount unlink: 2 'bar'
2015-11-25 09:26:49 arvados.arvados_fuse[21735] DEBUG: collection notify del <arvados_fuse.fusedir.UnsaveableC
ollection object at 0x7f3006218150> bar <arvados.arvfile.ArvadosFile object at 0x7f3006270f10>
2015-11-25 09:26:49 arvados.arvados_fuse[21735] DEBUG: del_entry on inode 5 with refcount 1
2015-11-25 09:26:49 arvados.arvados_fuse[21735] DEBUG: <arvados_fuse.fusedir.TmpCollectionDirectory object at
0x7f3006218190> invalidated collection record
2015-11-25 09:26:49 arvados.arvados_fuse[21735] DEBUG: arv-mount forget: inode 5 nlookup 1 ref_count 1
2015-11-25 09:26:49 arvados.arvados_fuse[21735] DEBUG: arv-mount forget: inode 4 nlookup 1 ref_count 1
2015-11-25 09:26:49 arvados.arvados_fuse[21735] DEBUG: arv-mount forget: inode 2 nlookup 1 ref_count 1
FAIL
Sent SIGTERM to 21748 (/home/peter/work/arvados/tmp/keep0.pid)
[keep0] 2015/11/25 09:26:49 caught signal: terminated
[keep0] 2015/11/25 09:26:49 keepstore exiting, pid 21748
Sent SIGTERM to 21756 (/home/peter/work/arvados/tmp/keep1.pid)
[keep1] 2015/11/25 09:26:49 caught signal: terminated
[keep1] 2015/11/25 09:26:49 keepstore exiting, pid 21756
```

```
=====
FAIL: test_tmp_snapshots (tests.test_tmp_collection.TmpCollectionTest)
-----
```

Traceback (most recent call last):

```
File "/home/peter/work/arvados/services/fuse/tests/integration_test.py", line 66, in wrapper
    return func(self, *args, **kwargs)
File "/home/peter/work/arvados/services/fuse/tests/test_tmp_collection.py", line 101, in test_tmp_snapshots
    self.pool_test(os.path.join(self.mnt, 'zzz'))
File "/home/peter/work/arvados/services/fuse/tests/integration_test.py", line 34, in pool_test
    (modName, clsName, '_' + funcName, args, kwargs))
File "/usr/lib/python2.7/multiprocessing/pool.py", line 244, in apply
    return self.apply_async(func, args, kwds).get()
File "/usr/lib/python2.7/multiprocessing/pool.py", line 558, in get
    raise self._value
```

```
AssertionError: Regexp didn't match: '^$' not found in u'. 37b51d194a7513e45b56f6524f2d51f2+3+Adfa8a6470676bd3
```

530c8a6594bd76f2dda806fe7056683a29 0:3:bar\n'

-----  
Ran 1 test in 3.007s

Here's a successful one:

```
[...]  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: arv-mount unlink: 2 'foo'  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: collection notify del <arvados_fuse.fusedir.UnsaveableC  
ollection object at 0x7f09dd93a150> foo <arvados.arvfile.ArvidosFile object at 0x7f09dd379c10>  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: del_entry on inode 3 with refcount 1  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: <arvados_fuse.fusedir.TmpCollectionDirectory object at  
0x7f09dd93a190> invalidated collection record  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: arv-mount forget: inode 3 nlookup 1 ref_count 1  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: arv-mount lookup: parent_inode 2 name '.arvados#collect  
ion' inode 4  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: arv-mount read 6 0 4096  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: arv-mount unlink: 2 'bar'  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: collection notify del <arvados_fuse.fusedir.UnsaveableC  
ollection object at 0x7f09dd93a150> bar <arvados.arvfile.ArvidosFile object at 0x7f09dc237f10>  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: del_entry on inode 5 with refcount 1  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: <arvados_fuse.fusedir.TmpCollectionDirectory object at  
0x7f09dd93a190> invalidated collection record  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: arv-mount forget: inode 5 nlookup 1 ref_count 1  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: arv-mount read 7 0 4096  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: arv-mount forget: inode 4 nlookup 1 ref_count 1  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: arv-mount lookup: parent_inode 2 name 'foo' not found  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: arv-mount create: 2 'foo' 100644  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: collection notify add <arvados_fuse.fusedir.UnsaveableC  
ollection object at 0x7f09dd93a150> foo <arvados.arvfile.ArvidosFile object at 0x7f09dd379f50>  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: <arvados_fuse.fusedir.TmpCollectionDirectory object at  
0x7f09dd93a190> invalidated collection record  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: collection notify mod <arvados_fuse.fusedir.UnsaveableC  
ollection object at 0x7f09dd93a150> foo (<arvados.arvfile.ArvidosFile object at 0x7f09dd379f50>, <arvados.arvf  
ile.ArvidosFile object at 0x7f09dd379f50>)  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: <arvados_fuse.fusedir.TmpCollectionDirectory object at  
0x7f09dd93a190> invalidated collection record  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: arv-mount write 8 0 3  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: collection notify write <arvados_fuse.fusedir.Unsaveabl  
eCollection object at 0x7f09dd93a150> foo (<arvados.arvfile.ArvidosFile object at 0x7f09dd379f50>, <arvados.ar  
vfile.ArvidosFile object at 0x7f09dd379f50>)  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: <arvados_fuse.fusedir.TmpCollectionDirectory object at  
0x7f09dd93a190> invalidated collection record  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: arv-mount lookup: parent_inode 2 name '.arvados#collect  
ion' inode 4  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: collection notify mod <arvados_fuse.fusedir.UnsaveableC  
ollection object at 0x7f09dd93a150> foo (<arvados.arvfile.ArvidosFile object at 0x7f09dd379f50>, <arvados.arvf  
ile.ArvidosFile object at 0x7f09dd379f50>)  
2015-11-25 09:35:43 arvados.arvados_fuse[22738] DEBUG: <arvados_fuse.fusedir.TmpCollectionDirectory object at  
0x7f09dd93a190> invalidated collection record  
[keep1] 2015/11/25 09:35:43 [[::1]:41523] PUT 37b51d194a7513e45b56f6524f2d51f2 0.000374s 200 86 "OK"  
[keep0] 2015/11/25 09:35:43 [[::1]:46603] PUT 37b51d194a7513e45b56f6524f2d51f2 0.000245s 200 86 "OK"  
.  
.  
.  
-----  
Ran 1 test in 3.500s
```

OK

**#21 - 11/25/2015 03:21 PM - Peter Amstutz**

It's working better with the no-cache trick in [9d80d31](#)

**#22 - 11/25/2015 03:33 PM - Peter Amstutz**

7751-mount-tmp looks good to me.

**#23 - 11/30/2015 03:00 PM - Tom Clegg**

7751-crunch-fuse-output @ [70111b9](#) lets you do stuff like this included example script:

```
#!/usr/bin/env python
```

```
import arvados
import arvados.crunch
import hashlib
import os

out = arvados.crunch.TaskOutputDir()

string = open(__file__).read()
with open(os.path.join(out.path, 'example.out'), 'w') as f:
    f.write(string)
with open(os.path.join(out.path, 'example.out.SHA1'), 'w') as f:
    f.write(hashlib.shal(string).hexdigest() + "\n")

arvados.current_task().set_output(out.manifest_text())
```

#### #24 - 11/30/2015 09:11 PM - Peter Amstutz

```
--volume=/tmp/crunch-job-1001/task/localhost.1.keep/tmp:/keep_tmp:ro
```

I don't think mounting the tmp directory as read-only is going to do what you want :-)

#### #25 - 12/01/2015 04:54 PM - Tom Clegg

Peter Amstutz wrote:

I don't think mounting the tmp directory as read-only is going to do what you want :-)

Indeed. Fixed, now at [9ce4db9](#)

#### #26 - 12/01/2015 06:29 PM - Peter Amstutz

Tom Clegg wrote:

Peter Amstutz wrote:

I don't think mounting the tmp directory as read-only is going to do what you want :-)

Indeed. Fixed, now at [9ce4db9](#)

Tested at it work now.

Rest of it looks good to me.

#### #27 - 12/02/2015 07:22 PM - Brett Smith

Merge of the new branch is blocked until we build and deploy new compute node images with the FUSE branch.

#### #28 - 12/02/2015 08:39 PM - Brett Smith

- Target version changed from 2015-12-02 sprint to 2015-12-16 sprint

- Story points changed from 2.0 to 0.0

#### #29 - 12/03/2015 04:55 PM - Tom Clegg

- Status changed from In Progress to Resolved

- % Done changed from 86 to 100

Applied in changeset arvados|commit:5590c9ac669f2d74858e6c994afe1a2e9df8d104.