2019-08-01 - Fedora Tech Meeting

Time/Place

This meeting is a hybrid teleconference and slack chat. Anyone is welcome to join...here's the info:

- Time: 11:00am Eastern Daylight Time US (UTC-4)
- Audio/Video Conference Link: https://lyrasis.zoom.us/my/fedora
    - Dial-in:
      +1 408 638 0968
      +1 646 876 9923
      +1 669 900 6833
      Meeting ID:
      812 835 3771
- Join fedora-project.slack.com on the "tech" channel

Attendees

Part 1:

1. Danny Bernstein
2. Andrew Woods ⭐
3. David Wilcox
4. Peter Winckles
5. Ben Cail
6. Aaron Birkland
7. Ben Pennell
8. Paul Cummins
9. Peter Eichman

Part 2

1. Danny Bernstein ⭐
2. Andrew Woods
3. Ben Cail
4. Aaron Birkland ⭐
5. Ben Pennell
6. Peter Eichman

Agenda

1. Announcements
    a. Docker options: Deployment Tooling
    b. Fedora 5.1.0 Release
2. Update on Fedora 6 Pilots (NLM, Docuteam, UWM)
    a. migration-utils work
        i. Configure "fedora4Client" to new implementation of the Fedora4Client.java interface
        ii. Writing to OCFL instead of Fedora4/5/6 API
3. Sprint Planning
    a. 6.0 Architecture Review
    b. Problems requiring design
        i. Transaction and Lock Management
            1. Transactions scope: what do we need to support?
                a. Individual Resources
                b. Groups of resources
                c. Containment hierarchies
            2. A Fedora transaction may span multiple HTTP Requests. Assuming that we do not want to commit anything to OCFL until the Fedora Transaction is committed, how do we maintain the state of open OCFL Sessions
                a. across requests?
                b. across instance reboots?
                c. across multiple horizontally scaled instances?
                d. How also do we ensure that two requests using the same transaction ID against the same OCFL object do not stomp on each other
            3. Globally accessible state (for horizontal scalability)
        ii. Caching/indexing strategy
            1. What caches and indexes do we need(ie in what layer(s)?)
                a. OCFL client
                b. Persistence Implementation (OCFL)
                c. Kernel Implementation
            2. Physical location of the cache (assuming we want to plan for horizontal scalability support)
                a. Cache per instance?
                    i. synchronizing changes across instance

        b. 1 Global cache/index?
4. OCFL implementation updates
    a. migration-utils pull-request
    b. Peter Winckles work-in-progress
        i. https://github.com/pwinckles/ocfl-java-parent
        ii. https://github.com/pwinckles/ocfl-java-api
        iii. https://github.com/pwinckles/ocfl-java-core
        iv. https://github.com/pwinckles/ocfl-java-filesystem
5. Your topic here...

Tickets

1. In Review

type	key	summary	assignee	reporter	priority	status	resolution	created	updated	due

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

2. Please squash a bug!

key	summary	type	created	updated	due	assignee	reporter	priority	status	resolution

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

3. Tickets resolved this week:

key	summary	type	created	updated	due	assignee	reporter	priority	status	resolution

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

4. Tickets created this week:

key	summary	type	created	updated	due	assignee	reporter	priority	status	resolution

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

Notes

Announcements

1. 5.1.0 release is out
    - Includes bugfix that allows import/export to work correctly
    - Testing of round-tripping is encouraged
2. Michael Klein has put out a new Docker image

Fedora 6 Pilots

General updates

1. Three pilot institutions
    - DocuTeam
    - Natl Library of Medicine
    - Univ of Wisconsin - Madison
2. It would be great to have addtional "partners"
    - Even if not as full "pilots"
3. Collecting requirements from pilot partners
4. Pilot effort will be running for the ~next year
5. Pilot meetings will be scheduled as needed
6. There is an open #fedora6-pilots Slack channel

OCFL Implementation

1. Peter W has put something together recently
2. Open questions around what the interface should look like
3. Currently supports PUT/GET... and most of the OCFL spec 🙂

F6 Architecture

1. Important for us to establish a shared understanding of transactions/locking/etc on how the architectural layers should interact
2. https://wiki.duraspace.org/display/FF/Fedora+2019+Architectural+Diagrams
    - HTTP Layer remains intact
    - Kernel Layer
        - Could potentially support swappable implementations
        - Session manages transactions/locking
    - Persistence Layer
        - Some features from prior ModeShape layer will move here
    - OCFL Layer
        - State may be maintained at different layers
3. Transactions and Locking
    - Currently, support txns across resources
    - Maybe scoping txns to a single OCFL object may make sense
    - If we want to scope txns across objects, that would have to be managed at the Fedora level... and the problem becomes more challenging
    - Proposal: scope txns to a single OCFL object
        - Primary use case: batch loader creating a complex resource
    - Do we want to consider architecting for cross-object txns?
        - Design the HTTP API to potentially support other scopes of txn, but implement for the single OCFL object case
        - Archive Groups may be a door for broader scope for txns
    - Would be nice if the txn-endpoint were advertised as a resource header... as opposed to a triple
    - Ideally not coupling txn-model to ldp-model
        - Bulk updates require cross-object txns
    - Do we need to maintain session state across clients?

...The conversation will be continuing at 1pm ET

Horizontal scalability:

- Aaron Birkland : sticky sessions is probably as far as we want to go; distributed transaction management is going to be very complicated.

migration-utils

Mike Durbin created migration-utils to facilitate 3-4 migration. Andrew has extended that project to migrate from Fedora 3 to OCFL.

Expectations for migration-utils:

1) Akubra

2) Legacy FS

3) Archival Export of Fedora 3

- Mike's code iterates the tree and migrates the resources.
- Andrew Woods has implemented an OCFL writer using Aaron Birkland 's OCFL go client.
- Results were promising using Peter Winckles 's data from UW-Madison
- Question: should we consider using Peter Winckles 's java client instead / in addition to?

It would be good to fix the travis.yml file to install Go and the ocfl client.

Meeting part II

Transactions

Action on transactions? Seems like we don't want to preclude arbitrary resources in transactions

Also may need to look at the transactions API. The existing transaction API isn't part of the Fedora spec. There is a draft of a new API, but it has no standing

Action: Peter Eichman and Ben Pennellgather opinions on transactions and APIs

Question: How much do we have to plan about distributed/multiple Fedora instances?

To what extent do we have to plan for this pathway?

Ben Cail: Not interested in multiple Fedora instances. Don't anticipate needing to scale, don't want to worry about concurrency. Would look at it if it came for free

Maybe that's a question that should go out to the community?

Don't want to get into transaction management.

State/locking

How much can we guarantee?

If we do a transaction involving bits and pieces of multiple OCFL objects, it is even possible to roll that back without direct support for these sorts of transactions in the OCFL client? Probably not.

Strawman: What if an OCFL client supported such transactions like this? How could that be implemented? Is it even desirable?

- Write file content to staging places as usual
    - If a failure/rollback happens here, we just have un-referenced staged content we can garbage collect later (GC)
- Maintain a database table that authoritatively OCFL metadata (that which gets written to inventory files)
- Upon "commit", copy files to the right location in OCFL
    - If this fails, then these copied files can be GC'd later. They aren't referenced by any inventory files, so as far as other OCFL clients are concerned, the incompletely-copied files are invisible. That being said, the OCFL objects they've been copied to are technically invalid until these files are removed.
- Commit (or roll back) that db table. Report transaction success if that succeeds
- Asynchronously, start writing the inventory files to OCFL that references the copied content.
    - If there's a failure here, that's OK. The authoritative DB still has the information in it to re-try until all the inventory files on the FS agree with the db.

TODO: Aaron Birklandflesh this out. Maybe bounce it by the OCFL community. Is an OCFL client that behaves that way "proper"?

Actions

- Danny Bernstein will reach out to Greg about DRASTIC test results
- Aaron Birkland to work with Andrew Woods to get the Go client working on travis.
- Aaron Birkland to look explore notion of OCFL client with database as authoritative metadata source + asynchronous writing of the inventory.json file
- Peter Eichman and maybe Ben Pennell to make recommendations re transaction side car specification.
-