

2012-02-29 Skype chat re Fedora + DuraCloud

[1:52:12 PM] Bill Branam: An interesting comment from Matt Zumwalt on the hydra-users list:

We have one project where we wanted to use Amazon S3 for delivery of streaming audio and video. To support that, we did not use "E" datastreams. Instead, we have an S3 xml datastream that contains the account id, bucket id, and file id needed to generate an authenticated S3 url or access/update the file. We overrode the "add file" behaviors on those objects to save files to S3 instead of Fedora and update the S3 xml accordingly.

[1:54:43 PM] Bill Branam: It would seem that we could do something similar to point Fedora objects to DuraCloud. That would potentially give us the opportunity to have Fedora store content in DuraCloud directly (rather than just be exported via CloudSync)

[1:59:24 PM] Daniel W. Davis: Chris and I have been chatting about the integration.

[2:02:35 PM] Daniel W. Davis: He was about to attack unoptimal behavior for handling the datastreams. This deserves factoring into the discussion, thanks.

[2:04:06 PM] Daniel W. Davis: We did want to get the DuraCloud user id added to the JMS message when possible. But this seems like a viable alternative.

[2:04:53 PM] Bill Branam: Of course, CloudSync would still be needed to get the foxml and non-content datastreams into Duracloud

[2:08:36 PM] Bill Branam: Andrew captured the need for the userId in messages here: <https://jira.duraspace.org/browse/DURACLOUD-628>

[2:09:18 PM] Bill Branam: It would probably make sense to link that issue to a jira in DfR at some point

[2:09:40 PM] Andrew Woods: I think Chris is on that linking.

[2:10:08 PM] Bradley Mclean: Hmm. Performance of S3 vs EBS an issue here or not?

[2:11:56 PM] Bill Branam: Brad, my guess is that performance is likely not a big issue in this case. As long as it's just the content files being stored directly, then we're likely to be OK. If we consider storing foxml directly, that's another story.

[2:12:17 PM] Bradley Mclean: That sounds right to me.

[2:37:11 PM] Andrew Woods: Dan & Chris: any new directions on the Fedora/DuraCloud integration (with the exception of the forementioned direct access)?

[2:54:54 PM] Daniel W. Davis: Sorry, I had to drop out for a short while to handle probate questions.

[2:55:22 PM] Andrew Woods: that happens

[2:57:49 PM] Daniel W. Davis: Re: The question. I don't know but that specific integration point is part of a new story/task Chris is writing since he finished his epic really fast (and below estimated story points). So we can pull in a new task.

[2:58:06 PM] Daniel W. Davis: This integration was not specifically on the backlog but should have been.

[2:59:02 PM] Daniel W. Davis: So, if its OK, I would propose we add it to the backlog, pull it in and work it.

[2:59:16 PM] Andrew Woods: so you guys were talking jira? not implementation?

[2:59:40 PM] Andrew Woods: I was just curious if any new ideas had sprouted on that front

[3:00:18 PM] Daniel W. Davis: Yeah, adding to Jira but Chris wants to work some issues regarding the datastream handling to enable connections to DuraCloud.

[3:00:35 PM] Daniel W. Davis: This description above will help the design.

[3:00:55 PM | Edited 3:02:09 PM] Daniel W. Davis: And if the design looks good he want to work it for this release.

[3:01:39 PM] Daniel W. Davis: It partially a Fedora mod that will make integration easier.

[3:02:58 PM] Daniel W. Davis: In our discussion yesterday, he took the action to write it up in Jira for the group to look at.

[3:36:03 PM] Chris Wilper: The relevant issue is <https://jira.duraspace.org/browse/FCREPO-748> (ability to authenticate to http-based storage when getting "E" datastreams). I had hacked something up for this a while ago, specifically to work with DuraCloud, but it never made it into trunk.

[3:37:46 PM] Chris Wilper: Having this would give Fedora the ability to provide the content of the original researcher data, which is stored in DuraCloud, directly through Fedora APIs, which is important if Fedora is going to be our policy enforcement point.

[3:38:29 PM] Andrew Woods: As I recall, it was a reasonable hack.

[3:41:03 PM] Chris Wilper: The Zumwalt approach gets around the need for that by having a stand-in datastream in Fedora. By "overriding the add files behavior" on those objects, it's not clear whether they made a change inside Fedora for that or (what I assumed), did it outside Fedora, somewhere in Hydra. I'm not sure what the advantage of the Zumwalt approach would be if we actually already had FCREPO-748.

[3:46:48 PM] Chris Wilper: The current behavior of OCS, fyi, is to use "R" (redirect) datastreams to reference the original researcher data. The problem with this is that when they're requested, Fedora only tells you the URL you can use to get them, and that URL is a DuraCloud URL that is only protected via basic DuraCloud capabilities. I've been assuming that most access **needs** to be mediated through Fedora for access control reasons.

[3:47:05 PM] Daniel W. Davis: I underestimated how much researchers peek at each others data.

[3:48:18 PM] Daniel W. Davis: There will likely be a lot of pre-publication controls researchers will want.

[3:49:33 PM] Bill Branam: Chris, it seems like FCREPO-748 combined with our approach to drop files in DuraCloud and then let OCS create the object in Fedora would do a good job of dealing with the need to provide access to those files, essentially GETs on DuraCloud content through Fedora.

[3:51:01 PM] Bill Branam: What the Zumwalt approach seems to offer is a strategy for allowing the content which is PUT into Fedora to land in DuraCloud storage directly

[3:53:43 PM] Bill Branam: My assumption on reading his comment was that the change to the add process was something they did within Fedora, to change the way Fedora stored content. I was wondering if this was done at the Akubra level, or just above

[3:54:16 PM] Daniel W. Davis: I bet we can find out.

[3:55:16 PM] Daniel W. Davis: On the surface it seems rather S3 specific.

[3:56:29 PM] Bill Branam: It does sound S3 specific, so we likely couldn't re-use the code, but perhaps the strategy

[3:56:59 PM] Chris Wilper: Yes, I'm curious. Was thinking about digging in on the mailing list where he posted that...I'm not finding the hydra-users list right off. I'm on hydra-tech, but where's hydra-users hosted?

[3:57:32 PM] Bill Branam: http://groups.google.com/group/hydra-users/browse_thread/thread/6f510fbf4bf9a01d

[3:57:35 PM] Chris Wilper: There are a few other "get it more directly into DuraCloud" approaches we've talked about too, which would be interesting to compare.

[3:57:43 PM] Daniel W. Davis: It seems like a good feature to be able to stream content through Fedora's APIs directly to DuraCloud. I thought we only avoided that approach to limit the scope for this grant.

[3:58:55 PM] Daniel W. Davis: So we could concentrate on the "slippery slope" of synching directly with DuraCloud.

[3:59:07 PM] Daniel W. Davis: From the user.

[3:59:50 PM] Chris Wilper: I think that feature is good in general for Fedora+DuraCloud, but I think it's good that our solution doesn't (currently) depend on needing to have it...I like the approach we've taken so far of leveraging lots of what we already have.

[4:02:04 PM] Andrew Woods: It is more than we likely want to bite off in this timeframe, but a DuraCloud/Fuse impl still seems attractive for the Fedora usecase as well as many others.

[4:03:01 PM] Daniel W. Davis: Andrew: Do you think write in Fuse is reliable enough.

[4:04:27 PM] Daniel W. Davis: Also, I would like to see a horizontally scalable method that does not overly rely on a cloud providers proprietary interfaces /services.

[4:07:43 PM] Daniel W. Davis: Regardless, I will take the action to write up the Fedora+DuraCloud ingest for the backlog (so as to not lose it for later).

[4:07:54 PM] Chris Wilper: Fuse itself is pretty reliable IMO. Of course it's slower than native filesystems in general, but the real problem is when you try to use it to front networked resources and don't put any caching in place. Then it hurts. Gluster <http://www.gluster.org/> is a good example of a Fuse-based

filesystem that is hz scalable.

[4:09:13 PM | Edited 4:09:24 PM] Daniel W. Davis: I figured that there would need to be a fast cache then an asynchronous write to DuraCloud for eventual consistency.

[4:09:25 PM] Chris Wilper: Andrew and I have talked about this a bit before too. Both a caching fuse-based filesystem "under" akubra-fs, and an caching akubra-duracloud implementation seem like promising technical routes.

[4:09:58 PM | Edited 4:11:08 PM] Bill Branam: For DfR we expect the majority of content to come in via the OCS route. But since we're considering DuraCloud to be the primary content store (where bit integrity checking happens), it would be nice to be able to push any files coming in through Fedora there more directly. The CloudSync method gets it there as well, but it takes longer and we do end up duplicating the storage space needed.

[4:11:22 PM] Daniel W. Davis: By expanding on the ideas of the OCS, it would be possible to deploy many fairly lightweight ingest pipelines in parallel informing everyone for eventual consistency.

[4:12:56 PM] Bill Branam: Dan, meaning that a PUT into Fedora would be picked up by OCS like a PUT into DuraCloud ?

[4:13:09 PM] Daniel W. Davis: Yep.

[4:14:13 PM] Bill Branam: I can see how that would be possible, but it seems a little round-about unless you're planning to do something else in the pipeline. [4:15:18 PM] Chris Wilper: Oh, that reminds me of an issue that came up when doing the first rev of OCS. I noticed that occasionally, when I got a message from DuraCloud JMS, if I immediately checked for the content, it wasn't there. That is expected, correct? IN other words, there is still no guarantee (except by waiting/polling) that the content is actually available by the time the message is sent.

[4:15:28 PM] Daniel W. Davis: Right, but notification is at least one immediate value. The main advantage that the cloud providers have is they parallelize the uploads.

[4:16:05 PM] Daniel W. Davis: If uploads have to go through a central server you will never get good horizontal scalability.

[4:16:47 PM] Bill Branam: Chris, that's correct, there is no current way to guarantee the file will be immediately available when the message comes in

[4:17:06 PM] Daniel W. Davis: Even DuraCloud may want to do a more complete job in characterizing the files.

[4:18:09 PM] Bill Branam: yes, file type analysis is on the DuraCloud roadmap, first step is just to verify/set the mimetype

[4:18:20 PM] Daniel W. Davis: Chris: IMO, handling asynchronicity will be a big thing going forward.

[4:19:05 PM] Daniel W. Davis: Most of this is future stuff anyway but I want to get it captured in one of the roadmaps, DuraCloud, Fedora or DfR.

[4:19:34 PM] Bill Branam: So for DfR, the direct route of "PUT to fedora and land in DuraCloud" is likely just an optimization of what we are currently planning. I agree with Chris that it's likely to provide a bigger gain for the Fedora+DuraCloud integration effort

[4:21:11 PM] Daniel W. Davis: Agile, working first — optimization later. But its cool to think where things need to go.

[4:21:28 PM] Chris Wilper: Yes. It seems like the best we can do is wait and poll today. Though it's notable that we can use much more efficient approaches against cloud storage for which there is a guarantee of immediate availability (amazon actually does this in certain regions). IN the end, it boils down to who is doing the caching.

[4:21:53 PM] Daniel W. Davis: Elliot Metsger keeps telling me that because I think too laterally and need to focus.

[4:23:32 PM] Daniel W. Davis: Chris: We actually have some issue in Fedora to smooth the messages when "content is expected to be there but is not available". That would help in a polling situation. Mostly an appropriate HTTP return code.

[4:23:49 PM] Daniel W. Davis: We put it in for supporting hierarchical storage.