DevMtg 2018-08-15

Developers Meeting on Weds, August 15, 2018

Today's Meeting Times

- DSpace Developers Meeting / Backlog Hour: 20:00 UTC in #duraspace IRC or #dev-mtg Slack channel (these two channels sync all conversations)
    - Please note that all meetings are publicly logged

Our IRC logging bot has been blocked from Freenode (as of July 27).

Discussion logs are no longer available at http://irclogs.duraspace.org/. As our current IRC log bot (based on PircBot) is unmaintained and doesn't align with Freenode policies (around requiring SASL authentication), Tim has reached out to https://botbot.me/ to see if they could log our #duraspace IRC channel. In the meantime, full logs of meeting discussions will be copied into the Wiki notes below.

Agenda

Quick Reminders

Friendly reminders of upcoming meetings, discussions etc

- DSpace 7 Working Group (2016-2023): Next meeting is tomorrow, Thurs, August 16 at 14:00 UTC
- DSpace Entities Working Group (2017-18): *This Working Group will be replaced by DSpace 7 Entities Working Group (2018-19)*
- DSpace Developer Show and Tell Meetings: Next Show & Tell will be August 28 at 15:00 UTC. The topic will be DSpace On DockerHub

Discussion Topics

If you have a topic you'd like to have added to the agenda, please just add it.

1. (*Ongoing Topic*) DSpace 7 Status Updates for this week.
    a. DSpace 7 Working Group (2016-2023) is where the work is taking place
    b. DSpace 7 Dev Status spreadsheet: https://docs.google.com/spreadsheets/d/18brPF7cZy_UKyj97Ta44UJg5Z8OwJGi7PLoPJVz-g3g/edit#gid=0
2. *(Ongoing Topic)* DSpace 6.x Status Updates for this week
    a. 6.4 will surely happen at some point, but no definitive plan or schedule at this time. Please continue to help move forward / merge PRs into the dspace-6.x branch, and we can continue to monitor when a 6.4 release makes sense.
3. Discussion topics / half-baked ideas (*Anything more to touch on with these?*)
    a. Bulk Operations Support Enhancements (from Mark H. Wood)
        i. Better support for bulk operations (in database layer), so that business logic doesn't need to know so much about the database layer. Specifically, perhaps a way to pass a callback into the database layer, to be applied iteratively to the results of a query.
        ii. Then, the database layer can handle batching, transaction boundaries, and other things that it should know about, and the business logic won't have to deal with them.
        iii. This is the result of thinking about a recent -tech posting from a site with half a million objects that needed checksum processing.
        iv. (This is almost an extension of the tabled topic below regarding DSpace Database Access, but a bit more specific in trying to simplify/improve upon how bulk operations are handled)
    b. Curation System Needs (from Terrence W Brady)
4. How to encourage / credit folks who do Code Reviews? (Tim Donohue)
    a. We have a lot of open PRs. As we know, the process for reviewing is very ad-hoc, sometimes encounters delays. If we can find ways to encourage/empower folks (even non-Committers if they know Java / Angular well) to do code reviews & be credited publicly...maybe we can speed up this process?
    b. Other brainstorms welcome!
5. Tickets, Pull Requests or Email threads/discussions requiring more attention? (*Please feel free to add any you wish to discuss under this topic*)

Tabled Topics

These topics are ones we've touched on in the past and likely need to revisit (with other interested parties). If a topic below is of interest to you, say something and we'll promote it to an agenda topic!

1. Management of database connections for DSpace going forward (7.0 and beyond). What behavior is ideal? Also see notes at DSpace Database Access
    a. In DSpace 5, each "Context" established a new DB connection. Context then committed or aborted the connection after it was done (based on results of that request). Context could also be shared between methods if a single transaction needed to perform actions across multiple methods.
    b. In DSpace 6, Hibernate manages the DB connection pool. Each **thread** grabs a Connection from the pool. This means two Context objects could use the same Connection (if they are in the same thread). In other words, code can no longer assume each `new Context()` is treated as a new database transaction.

i. Should we be making use of `SessionFactory.openSession()` for READ-ONLY Contexts (or any change of Context state) to ensure we are creating a new Connection (and not simply modifying the state of an existing one)? Currently we always use `SessionFactory.getCurrentSession()` in HibernateDBConnection, which doesn't guarantee a new connection: https://github.com/DSpace/DSpace/blob/dspace-6_x/dspace-api/src/main/java/org/dspace/core/HibernateDBConnection.java

Ticket Summaries

1. Help us test / code review! These are tickets needing code review/testing and flagged for a future release (ordered by release & priority)

key	summary	type	created	updated	assignee	reporter	priority	status	fixversions
Unable to locate Jira server for this macro. It may be due to Application Link configuration.									

2. Newly created tickets this week:

key	summary	type	created	assignee	reporter	priority	status
Unable to locate Jira server for this macro. It may be due to Application Link configuration.							

3. Old, unresolved tickets with activity this week:

key	summary	type	created	updated	assignee	reporter	priority	status
Unable to locate Jira server for this macro. It may be due to Application Link configuration.								

4. Tickets resolved this week:

key	summary	type	created	assignee	reporter	priority	status	resolution
Unable to locate Jira server for this macro. It may be due to Application Link configuration.								

5. Tickets requiring review. This is the JIRA Backlog of "Received" tickets:

key	summary	type	created	updated	assignee	reporter	priority
Unable to locate Jira server for this macro. It may be due to Application Link configuration.							

Meeting Notes

Meeting Transcript (IRC Bot is not working)

- ~~IRC Transcript is available at - http://irclogs.duraspace.org/index.php?date=2018-08-15~~

Log from #dev-mtg Slack (All times are CDT)

```
Tim Donohue [2:50 PM]
@here: Reminder that the DSpace DevMtg starts at the top of the hour (~10mins).  Rough agenda at https://wiki.
duraspace.org/display/DSPACE/DevMtg+2018-08-15


Tim Donohue [3:00 PM]
@here: Ok, it's DevMtg time.  Let's do a quick roll call to see who all is here.

Mark Wood [3:01 PM]
Here!

James Creel [3:01 PM]
As luck would have it, Texas Digital Library is running a DSpace users group meeting simultaneously - I will be
engaged with that, but watch the traffic here out of the corner of my eye.

Tim Donohue [3:02 PM]
Aha, no worries, @jcreel256. :wink:
Looks like we are a small crew today, but I'll go ahead and get the meeting started.  If more people don't join
into discussion, we can always wrap up early, or discuss specific tickets/PRs of interest (as needed)
I don't have any significant updates on DSpace 7 to share today.  We've had a lot of vacations in August, so it
is expected to be a slow month (with things picking up in late Aug / early Sept).
On the DSpace 6.x front, there's not much going on either....other than we still have PRs against 6.x that have
been submitted.  So, we can always use help reviewing/testing bug fixes, so we can see what is ready to go in
an (eventual) 6.4.
No exact timelines on a 6.4 though yet
So, that's a quick run through reminders & our usual ongoing topics.  Any questions/comments before we simply
dive into other discussion topics?

Mark Wood [3:06 PM]
No

Tim Donohue [3:07 PM]
Ok, moving along. Under #3 (a), I see you've linked in a new wiki page, @mwood.  https://wiki.duraspace.org
/display/~mwood/Bulk+Operations+in+DSpace
I hadn't had a chance to look at this yet (looking now).  Any updates/thoughts to share here?

Mark Wood [3:08 PM]
No, I just dashed down a few thoughts as a starting point.  About 15 minutes ago. :-/
But it gives a lasting place for accumulating ideas and critiques.

Tim Donohue [3:09 PM]
:+1:
The very last line here is definitely interesting...considering whether we should find a way to share code
/concepts between our ideas of "bulk operations" vs. "curation tasks".  It seems like an opportunity to
potentially simplify

Mark Wood [3:10 PM]
Comments would be welcome.
Hm, interesting.  I was thinking more along the line that the curation framework could *use* the bulk-operation
support, and might be a good place to pilot the idea.

Tim Donohue [3:12 PM]
I'm trying to determine what to do with these topics under #3 (as well).  Do we keep them on the weekly
agenda....move them down into "Tabled Topics" until we are ready to discuss again in more detail?  Is there
effort folks want to put into this now?

Pablo Prieto [3:12 PM]
Hi all!
```

Tim Donohue [3:12 PM]
@mwood: yes, that's essentially what I mean...hadn't though enough as to how curation tasks & bulk-operation would work together...just noting that it seems like they share a lot of concepts, and therefore could share code
@mwood: also, as I know you mentioned that email thread on -tech about checksum checking being a reason for this bulk operations brainstorm...it seems we already have a possible ticket & fix: https://jira.duraspace.org/browse/DS-3975
(That last point is a sidenote...we still need to discuss how to do this better in the future)
Just wanted to close the loop here on the fact that the checksum stability issues (which brought about this broader discussion) have had their own specific analysis & now a PR

Pablo Prieto [3:16 PM]
A noob question here. Do these operations run in series or is there any parallelism / concurrency ?

Mark Wood [3:16 PM]
IIRC that one is a special case: an operation that can be done entirely by the DBMS. We should keep our eyes open for such cases, because the efficiency there is quite attractive.
Since these are primarily operations against the database, it may be better if they are *not* parallel in any way.

Pablo Prieto [3:17 PM]
Ok

Tim Donohue [3:17 PM]
@Pablo Prieto: most (possibly all) of these run in series, as often times bulk operations are transactional.

Pablo Prieto [3:17 PM]
Oh, I understand.
Most servers are multi-core. There could be some performance gains if that is somehow used.
Although it could also consume all processing resources.

Mark Wood [3:19 PM]
We can use that hardware parallelism up in the UI layer, since we have potentially many concurrent users mostly doing *different* things.

Pablo Prieto [3:19 PM]
Ok

Tim Donohue [3:20 PM]
yes, often these bulk operations / curation tasks run in the background...so, they don't affect the UI performance/activities (ideally).

Mark Wood [3:20 PM]
The DBMS can also make good use of parallel execution, since it has a good understanding of what it has to do.
It's in the middle that parallelism may not be the best thing.

Pablo Prieto [3:21 PM]
But every operation references a unique object, right?

Tim Donohue [3:23 PM]
I'm not sure that's an easy question to answer in a generic way. Curation Tasks do tend to perform operations per object. But, we have other forms of bulk operations that don't necessarily reference individual options (like Bulk Metadata Editing, which is a single operation across many objects)

Pablo Prieto [3:23 PM]
Oh, I see.

Tim Donohue [3:24 PM]
So, while Curation Tasks & Bulk Operations are very related...we have several different types of bulk operations, and unfortunately the code behind them isn't always "shared code". So, individual operations may do things slightly differently
And I think that :point_up: is what @mwood is pointing out here, if I'm not mistaken

Mark Wood [3:25 PM]
Perhaps we can at least break down the set of all bulk operations into a small number of subsets that are internally similar, and find ways to do each type of bulk operation better.

Pablo Prieto [3:25 PM]
"An ORM query which matches a large quantity of objects will eat up large amounts of memory with a huge list of objects (or trees of objects!)" <-- Maybe this can be adresed if only the objecty IDs are queried and then,

each object be loaded into memory when used and then discarded.

@mwood Yes

Tim Donohue [3:26 PM]

@mwood yes, that likely would be helpful here. Mabye that's a list to start on that wiki page

I think this also does relate back to some of the discussions (now in "Tabled Topics") around Database Access (https://wiki.duraspace.org/display/DSPACE/DSpace+Database+Access). As, at least some of these bulk operations may be able to use Hibernate more efficiently

So, in a way, Bulk Operations, Curation Task, & Database Access topics all have overlaps here...they are different perspectives on similar issues

Mark Wood [3:28 PM]

The pattern I thought I saw was that we tend to do gigantic, expensive operations in the same way we do small, cheap ones, and that causes problems.

Tim Donohue [3:29 PM]

I guess the question here is...what do we do with these topics? Do I keep these on the weekly agenda? Is there a way to better summarize / link them together to move towards a "proposal" for specific changes?

Mark Wood [3:29 PM]

The design of contemporary languages tends to encourage this pattern, by making it easy to ignore scaling issues.

If a topic hasn't had any activity for a couple of weeks, beyond "nothing much to report," then it should move, I guesss.

Tim Donohue [3:31 PM]

@mwood: I agree that we do have that pattern. It manifested pretty significantly in the early pre-6.0 efforts... we had to adopt new patterns quickly to fix it (in Hibernate). Though, we haven't taken that step back now to analyze if there's a "better way" we could be doing this

I think this is an important topic, and it's one I'd like someone(s) to put some thought into. It almost sounds like it could make for an interesting "mini-working group" / "team" that could dig in deeper & report back.

For me though, I'd likely only be able to be supportive of this effort...I don't have much time to speak of right now to dig deeply here

Pablo Prieto [3:33 PM]

@tdonohue I agree with the notion that this is important. If this pattern doesn't scale well, and every institution is heading towards "huge repositories", this should be addressed. (edited)

Mark Wood [3:34 PM]

I never heard of a repository that shrank over time, so we are all heading towards huge repositories, though perhaps very slowly at some.

Pablo Prieto [3:35 PM]

Just to have an idea. What size do we consider huge?

Mark Wood [3:36 PM]

As I recall, the repository where they had the problem that set me to thinking about these issues had around half a million objects. Their experience seems to put them well beyond the lower bound of "huge".

Pablo Prieto [3:37 PM]

:astonished:

ok

Tim Donohue [3:39 PM]

@mwood: yes, but I think the issues you are referencing there are actually fixed/described in https://jira.duraspace.org/browse/DS-3975 I think that scenario was caused by bad code in 6.x...where a database query in 5.x was replaced by a massive Java loop through objects.

Mark Wood [3:40 PM]

So we *do* have at least two subsets: things that can (and should?) be done in the DBMS, and things that require processing in DSpace.

Tim Donohue [3:41 PM]

Yes, I think so.

Mark Wood [3:44 PM]

I've linked that issue as an example.

Tim Donohue [3:44 PM]

Sounds good.

```
I'm still not exactly how to link together Bulk Operations with Curation Tasks & past Database Access
discussions....but it seems like Bulk Operations encompasses both.  Curation Tasks are those "things that
require processing in DSpace" (object by object).
The past Database Access discussions seem to have a lot of overlap here...as they talked both about how to
process things object by object (e.g. for Curation Tasks) as well as how to build better HQL / Hibernate
queries (for DBMS bulk processing)
In any case, I think these three topics belong somewhat together...and I'll see if I can figure out a way to
describe it better / link them better in the Agenda

Mark Wood [3:48 PM]
Thank you.

Tim Donohue [3:48 PM]
So, moving along here...we only have 10mins left, but I want to touch briefly on topic #4 in the agenda.  It's
something that could use more discussion in future weeks
Simply put, as noted in the agenda, We have a TON of open PRs. https://github.com/DSpace/DSpace/pulls
We don't really have a formalized process for reviewing them in a timely basis.  We depend on folks picking
things up, chipping in , or asking for reviews (sometimes several times)
And I'd like to find a way to both...formalize a review process...and also, *credit folks* who do reviews (and
do them regularly / well)

Mark Wood [3:50 PM]
We used to do scheduled PR triage sessions, but needed the time for other things IIRC.

Tim Donohue [3:51 PM]
@mwood: yes, though I heard a lot of feedback about it being too many meetings, especially with various other
working groups (DSpace 7, Entities, etc)

Mark Wood [3:52 PM]
I would do more reviews if I felt that I understood the problems well enough to understand and test the
solutions.
Yes, we do have a lot of meetings.

Tim Donohue [3:53 PM]
So, I admit, I don't have a ton of great ideas here yet.  We could form a "Code Review Working Group" (and
either schedule meetings or organize ourselves virtually / hold ourselves accountable through other means)
We could start to highlight folks who did reviews/testing/commenting on GitHub in every release's Release Notes
(presuming I can get that list out of GitHub easily)
I'm also open to ideas/suggestions here on what might work...and we could even try something for a while, and
change direction if it doesn't work :wink:

Mark Wood [3:54 PM]
Maybe a script could send out a list of "10 most doomed PRs" weekly or something.
I did something like that to keep me aware of Jira issues.

Tim Donohue [3:55 PM]
@mwood: possibly, I'm just worried still that if no one is "accountable" / taking ownership of reviewing PRs,
that email itself might just get ignored.

Mark Wood [3:55 PM]
True

Pablo Prieto [3:55 PM]
If each dev could review some specific area that he's comfortable with, would there be too many devs
overlapping this "areas"? (edited)

Tim Donohue [3:56 PM]
I try my best to keep myself somewhat accountable...but there's way too much coming in the door for me to even
attempt to keep up with or forward along to others to review
@Pablo Prieto: I don't know, to be honest :slightly_smiling_face:  We'd have to come up with a list of "areas",
and then look to get folks to "sign up" to review PRs in that area...and also have someone identify which
"area" a PR belongs to (maybe tag/label it appropriately)
That said, that could work...and I could simply tag PRs to a specific "area"
It's much more scalable for me to tag every PR coming in the door than to try and review them all :wink:

Mark Wood [3:59 PM]
There are some areas where we have very few developers.  Oracle support is very thin right now.  The Hibernate
work was mostly done by one person who seems to be busy with other work now.

Tim Donohue [3:59 PM]
@mwood: true, but....I'm now wondering myself if listing out these "areas" of support would be useful in
```

```
showing those gaps more easily.

Mark Wood [4:00 PM]
JSPUI seems to be rare among developers' sites.

Pablo Prieto [4:00 PM]
Well, maybe there should be a policy of at least two devs en each area. I any of those decides to leave, go on
vacation or something, at least the know-how is non-exclusive.

Tim Donohue [4:00 PM]
I.e. if we had a big list of "areas" we could say... Look we don't have Oracle reviewers currently, if you
could help in that area, then PRs for Oracle would move along more quickly.
@Pablo Prieto: yes, I'd actually say the ideal is ~3 per area... we won't get to that initially, perhaps, but
I'd rather have plenty of backup, etc

Mark Wood [4:02 PM]
It is difficult to apply manpower policies in a volunteer organization.  Maybe we could work through the
steering group, which has a lot of people who could make learning to support a specific area part of the job of
a subordinate who happens to be a DSpace developer.

Tim Donohue [4:03 PM]
I'm realizing we are at the top of the hour...so, we'll likely need to wrap this up.  But, perhaps the next
step is to start to identify "reviewer/tester areas of concentration"....and then see if we can get support for
this concept

Mark Wood [4:04 PM]
That sounds well.

Pablo Prieto [4:04 PM]
All right

Tim Donohue [4:04 PM]
@mwood: right, agreed.  Sometimes the best way (I find) to get that manpower to appear is to show where the
gaps are.  Hence, if we show that we have no Oracle Reviewers, I may be able to bring that to Steering (and
others) and say "we need your help if you use Oracle"
So, I'll see if I can move this idea forward a bit in the next week or so. I'll start with a simple wiki page.
Thanks for the brainstorms/ideas @Pablo Prieto and@mwood !

Pablo Prieto [4:05 PM]
Great! Thank you!

Tim Donohue [4:05 PM]
And with that, we'll wrap up today's meeting.  Thanks again, and we'll talk again next week, Aug 22 at 15UTC!

Mark Wood [4:06 PM]
Thanks, all.
```