

2019-04-30 - VIVO Development IG

Date

30 Apr 2019

Call-in Information

Time: 11:00 am, Eastern Daylight Time (New York, GMT-04:00)

To join the online meeting:

- Go to: <https://duraspace.zoom.us/j/823948749>
- Or iPhone one-tap :
 - US: +14086380968,,823948749# or +16468769923,,823948749#
- Or Telephone:
 - Dial(for higher quality, dial a number based on your current location):
 - US: +1 408 638 0968 or +1 646 876 9923 or +1 669 900 6833
 - Meeting ID: 823 948 749
- International numbers available: https://duraspace.zoom.us/join?m=Qy8de-kt6W4fMMDQCAV_3qfH1W-lxAo5

Slack

- <https://vivo-project.slack.com>
 - Self-register at: <http://bit.ly/vivo-slack>

Attendees

[blocked URL](#) Indicating note-taker

1. [Don Elsborg](#)
2. [Huda Khan](#) ★
3. [Harry Thakkar](#)
4. [Brian Lowe](#)
5. [James Silas Creel](#)
6. [William Welling](#)
7. [Benjamin Gross](#)
8. [Mike Conlon](#)
9. [Quinn Hart](#)

Agenda

1. Reflection on TAMU Scholars Demo
2. Other topics

a.	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.

b. VIVO-Docker2 [pull-request and approach](#)

Tickets

1. Status of In-Review tickets

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. Received

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.

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

3. Bugs (1.11)

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.

Notes

Draft notes in Google-Doc

- Discussion of TAMU demo
 - Questions? Comments?
 - Brian: interested in Google Scholars and JavaScript problem. Wasn't on radar screen earlier. Is Google Scholars typical of indexers in this case?
 - Experience difficulties with DSpace Angular UI. Several attempts at stack for DSpaceUI and one requirement was that it was indexed by Google Scholars. Wanted it verified - Google Scholars contact (Anurag) indicated won't be indexed and has to be server-side rendered or isomorphic JavaScript that had server-side rendering
 - A lot of search indices do seem to run some amount of client-side code so things may have changed since
 - Don: Separate bot used by Google Scholars?
 - Yes. Seems like probably still the case - had some content unintentionally indexed by Google Scholars but had to go through them and not regular Google process to get that content 'unindexed'
 - Normal Google Scholar will run ES5 but most browsers don't run ES6 (?did I get this right)
 - Benjamin: directive for new site or also old?
 - For new (old was customized VIVO)
 - Spring <-> Elasticsearch compatibility: considering making TAMU scholars configurable with either Elasticsearch or Solr (thank you spelling person)
 - Don: GraphQL thoughts?
 - Great idea but pros/cons. Like the idea of efficiency of responses. Tell it schema of response and you get exactly that back. Misconception: get it all for free. Have to write a resolver for aggregation you wish to choose, seems like writing your endpoint for aggregation. Downside: lose REST since everything is a POST.
 - Don has a follow up: if business comes up and says they want to add a set of new fields into Scholars where it wasn't part of the original spec. How easy is it to do that in your design?
 - From ETL to process to end user interface
 - Current process: data expert updates ontology if necessary. Add data through VIVO. Scholars has to edit the JAVA model (responsible for translating between ontology and index) so that field is added. Proof of concept that exposes this through a UI.

- Dynamic Solr schema - caveats: Solr in schemaless mode sets type of field based on the first data that gets pushed to it unless the type is explicitly declared. Doesn't allow data type changes.
- Don: YAML with fields - wouldn't have to compile any JAVA code? (livin' the dream)
 - Yes. (With the UI proof of concept)
 - Once the API for updating Solr documents, would be happy to demo that. Proof of concept is too proof-of-concepty currently but may be good to demo later.
- Benjamin: If someone isn't a Spring expert, how could we deploy it?
 - One of the requirements to deploy that along with VIVO
 - Using TDB (although SDB still supported)
 - Have this service be external
 - SDB: set up URL and username/password
 - Have option to index: this is the entire index based on the triplestore. Takes 45 minutes to reindex currently. This is time to index their triple store. Counts are on the index of their TAMU site.
 - Have to set up JAVA container and Node server. Don't need to do anything with Spring.
 - Default index is very sparse (VIVO's own) whereas TAMU's index is very dense (almost all the triplestore is indexed)
 - Uses CRON descriptor (string) but in Spring's internal scheduler. Runs on a separate thread (not container's main thread) based on CRON. Indexer optimized with multi-threading.
 - Don: Listeners on Jena to re-index based on changes to triplestore?
 - No. VIVO's own code base: it's not really "listening" - calls method to model listener change on listener.
 - Library RDF Delta: patch server. Logs diff of all RDF transactions. Updates/backups/events you want to take action on. Good candidate for making VIVO a microservice participant so it's providing events through this RDF patch server that can then be consumed by other services.
 - And not a major overhaul of code. Requirement for spinning up the patch server.
 - [afs.github.io/rdf-delta](https://github.com/afsoffice/rdf-delta)
 - Benjamin: not linked data but enables sharing of data
 - Don: Is there a BlazeGraph counterpart?
 - Don't know. Perhaps options with Fuseki but they are not using Fuseki
- HUda - linked data notifications area, can this somehow piggy back off of RDF delta? Also, how do you granularity control what changes are processing. How do you know if a certain subject is modified. So if you're more interested in an entity vs. every single triple.
- TAMU - patch server serves all diffs of 3store, consume all, do inferences of the patch, filter, created messages, etc. Apache RDF delta might have messaging built in. nb. Probably not an apache project yet, but had origins in apache
- Huda - indexing in general - there's an indexing thread that is triggered, it's not triggered at 3store level but at code above it
- Tamu - yes - index service in thread that listens to model listener, it then queues up change and applies them so it's not bottleneck. This is invoked anytime the 3store makes a change. It makes sense that you have an in code app layer
- Huda - when updates to interface there's a collection of trips that are applied. This was to address real time edits to interface. So it hangs and waits to add triples to the queue.
- Don - would shapes have a role?
- Huda - might, should somehow connect to what you wait for before you index. Delta has the stream, then need code to determine what is required for change. So there can have shape to determine what is a person or other object.
- Tamu - indexing root models might work, but if some detail is being changed how to know if root model has to change eg if course changes how to know to filter on all the relationships to what is changed.
- Don: React and Angular. Angular seems like a full stack on the client side. Thoughts on React vs Angular?
 - Both serve their purpose. Angular is a framework. Have to use its API in coding Angular: components, services, directives, decorators, modules.
 - React similar: properties passed between components, inheritance.
 - Dislike: doesn't like multiple languages in one file.
 - Angular decouples using MVC approach
 - Don: DotJS library issues with maintaining state between Angular and DotJS templates
 - Resolved by re-templating by broadcast of changed data
 - TAMU: DotJS was second choice, probably moving back to a different templating system (e.g. mustache) (handlebars is a superset of mustache which makes somewhat conceptual sense? Although handlebars seem to be a TYPE of mustache in non-tech speak)
 - If something is being updated, the researcher would like to know about it, but on the other hand, page refresh would get latest info
- Don: Where are aggregations happening?
 - Solr supports them but not in an elegant way.
 - Ad hoc way. Index nested object in same index. Have to know that nested object exists there to query it.
 - Spring did not support it well.
 - Had a lot of nested objects - have lots of relationships you want to facet by that are not on the entity itself
 - Serializer: nested flattened maps out of Solr document into a nice API response
 - Elasticsearch does nesting well
 - Don: Ended up with redundant documents but that is fast. Not sure what the right way to do this is.
 - TAMU: Do want the things you want to filter/facet against within the index of the entity you're looking at.
 - GraphQL just aggregating won't provide that type of faceting or filtering
 - Don: Have DSL language (painless? paneless?) with administrative features to help with ETL
 - TAMU: Decided to flatten nested object but only bring in properties for filtering and faceting - so only partial nested document for other Solr documents
 - Others may want to filter/search on different things
 - Excited by dynamic Solr schemas using precompilation at run time
 - JAVA recompilation at run-time very well-supported and one of the main objectives of the JVM when it came out
- Benjamin: Duke Scholars group would be interested
 - Don: Yes. Shared it on Slack
 - TAMU: meeting with Duke developers on Friday to see if collaboration is possible
 - Let's look at pull requests and tickets stuff!

- Harry: In tech stack, how important is modularization too? How do you handle local development? Docker?
 - TAMU: separated back-end from front-end. Back-end not as modular as it can be - perhaps could be split into multiple services. First pass to try and satisfy core set of requirements to create UI on top of VIVO.
 - Front-end with Angular: can create modules and isolated in other apps. Routing is done lazily: only get HTML and JavaScript for the section you are looking at.
 - Using Docker. Builds solr and 6 cores. Eventually use kubernetes ... long term. Eventually use chef to match production cookbook. For dev just docker and java. Node uses pm2(?)
 - Harry - comes to meetings from devops view. Prod evol team goal was highly modularized platforms. So Dukes current impl is complete Kubernetes and docker-compose. So can start the whole stack with a single click.
 - TAMU - yes - that is the same method they want. Current is [Chef](#), but will do kubernetes with Rancher. Re: modularization, will have option to place all components on one or multiple servers. Can deploy each app, node, vivo, solr, etc into their own node containers.
- Benjamin - Jira - new issues raised any discussion?
 - Interesting one from Graham - UI one. Ben tried to build it but
 - Ben submitted a PR related to his IDE removing white space from code. It's a big PR but only whitespace. He is vested in this PR.
 - TAMU - can always filter whitespace from git
 - Andrew submitted separate issue for configuration for checkstyle (?) for enforcing code styles at deployment level. Anyone else do this <crickets>
 - PR Vitro - 111 - sparql code vulnerability. Ben hasn't tried this, but if people have time. This is a good one to look at.
 - PR vitro 106 - Hudao PR, might need tinkering. Ensures people that need to edit a record have capability to edit the record. The current method is a crude work around.

Actions

- Unable to locate Jira server for this macro. It may be due to Application Link configuration. - [Mike Conlon](#), can you give this one a

Previous Actions

-