Yale ArchivesSpace Performance Analysis Report

August 11, 2015

In August 2015 Yale University Library (YUL) contracted Hudson Molonglo (HM) to conduct a second investigation into remaining significant performance issues with operations on certain records in the YUL production deployment of ArchivesSpace. This report summarizes the methodology and results of the analysis undertaken by HM and provides recommendations based on the results.

The analysis was undertaken by Mark Triggs and Payten Giles with support from James Bullen. The inputs were materials made available by YUL including example records, configuration files, log files and database dumps.

Using the materials to establish a test environment, we were able to replicate the poor performance on the records reported by YUL. We were then able to identify a number of areas that were contributing to the poor performance, and to provide remedies in each case.

The remainder of this report has the following sections:

Resource record performance

Resource records with large numbers of agent and/or subject links perform slowly when rendering the “edit” screen. For example, the Frederick R Koch collection record took minutes to load in our testing. The breakdown of the load time was:

- ArchivesSpace backend - 6 seconds to fetch the record from the database and produce its JSON representation.
- ArchivesSpace frontend - 16 seconds to produce the HTML page for the edit form.
- User web browser - 2+ minutes to render the HTML page (which was around 10MB of HTML content) and to run the JavaScript required to initialize the edit form.

These poor load times stem from the fact that some records have many more linked agents and subjects than anticipated when the forms for linked records were originally designed. The current interface shows all linked records in an editable state and allows them to be dragged and dropped to reorder them. This works well for tens of linked records, but breaks down in both performance and usability when managing hundreds of linked records.

Given the potential for large numbers of linked records, a better approach would be to show the linked records in a more compact format, allowing the user to “expand” a particular entry to edit it. This removes the need for fetching and displaying all record data up front, and will allow the load times to be drastically reduced. Usability should also be improved, as the more compact display will allow more linked records to be seen at a glance. Figure 1 shows a mockup of how the modified interface might appear.

We also noticed that a large amount of time was spent rendering the indicator dots in the record navigation pane (see Figure 2). These are intended to give an idea of the size of the record, but with over 500 linked records, rendering these dots become prohibitively expensive, and the usability also suffers. We suggest changing this display to show a number to indicate the count of linked records, rather than one dot per record.

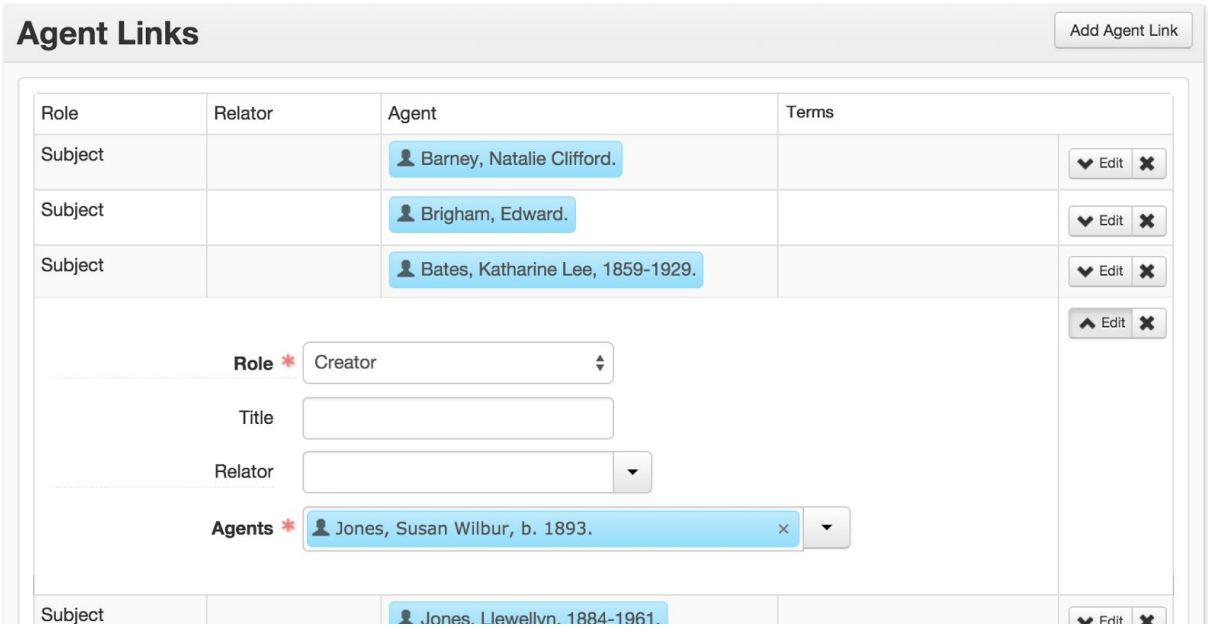

Figure 1: Compact linked record display

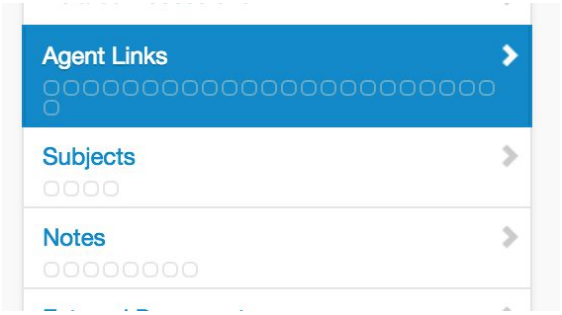

Figure 2: Subrecord Navigation

Accession record performance

Upon testing accession records we found it quite common for both the view and edit screens for an accession to load slowly. In this case, the slowness was attributable to the ArchivesSpace backend: in particular, the code that shows the events that are linked to the accession, and the agents that are in turn linked to those events. Although there are generally only a small number of events linked to an accession, fetching the entire agent record for each event is a relatively expensive operation, and this caused the record load times to exceed ten seconds in our testing.

A dramatic performance improvement could be obtained by querying the Solr search indexes for event information, rather than loading everything from the ArchivesSpace database with each request. This is consistent with the way linked records are shown in other parts of ArchivesSpace, and would be relatively straightforward to implement.

A secondary improvement would be to enhance the way that resolving records works in ArchivesSpace, to allow it to be more fine-grained. In ArchivesSpace "resolving" refers to the act of taking the URL of a linked record and fetching its metadata from the database. For example, when fetching an accession record via the API, a user might also want to obtain information about the subjects that are linked to that accession. By passing a special *resolve* parameter to the ArchivesSpace API, it's possible to fetch the record for the accession *and* its linked subject records in a single request--efficiently accessing the group of related records.

The ArchivesSpace frontend makes extensive use of resolving when displaying records. Currently resolving is "all or nothing" in the sense that resolving a record pulls back all information about that record, whether or not the user needs it. In the case of events and their agents, where the agent is complex and may have many notes and subrecords, this resolving process can be quite slow.

This situation could be greatly improved by allowing the ArchivesSpace frontend more fine-grained control over the information returned during resolving. For example, the "Agent Links" display only needs the name and type of each agent; it doesn't need extended agent information such as notes, dates, contacts, and so on. Currently the frontend pays the cost of fetching this superfluous information, but a small change to the resolving system would allow it greater control and better performance.

Recommendations

- **Rework the edit forms to show linked agents and subjects in a more compact way.** This will eliminate the need to display one edit form per linked record from the outset, and will drastically reduce the overall load time for edit pages, while enhancing usability.
- **Replace the subrecord navigation "dots" with numbers.** This will reduce the time spent rendering each edit page, and will make the feature more useful for large records.
- **Modify the display of linked events to be driven by the Solr indexes instead of the ArchivesSpace database.**
- **Review the use of "resolving" when displaying record forms. Ensure that we're not resolving records unnecessarily.**
- **Enhance the resolver system to be more flexible.** This will allow the ArchivesSpace frontend to limit the amount of information it pulls back on each record, giving more fine-grained control over performance.
- **Ensure that record forms are only loaded once when editing records.** We noticed a bug where sometimes records would be loaded twice during editing, which makes them twice as slow.
- **Further profile JavaScript once these enhancements have been made.** It would be beneficial to perform further JavaScript profiling once these known issues have been addressed.

Appendix 1: Indexer Issue

It was reported that users have been experiencing delays or omissions of data in search results--when searching for top container barcodes the expected results are not returned until several days after the change was made. In investigating this issue, we discovered a bug in the Container Management plugin whereby records linked to a top container are not reindexed after a top container is changed via a bulk operation. This would mean that any linked record would not be discoverable when searching with the barcode until the top container was saved by a change through the “Edit”-then-“Save” workflow.

We’ve prepared a new release of the container_management plugin will resolve this issue: https://github.com/hudmol/container_management/releases/tag/1.0.4