Fedora Statistics

Since Fedora objects will typically come from disseminations (without the .jpg or .gif extensions), we will have to take care to collect meaningful statistics. We can get detailed info on datastream/disseminator requests via the Tomcat logs.

Rob Chavez on statistics collection at Tufts

We've done some minimal reporting of this nature at Tufts using AWStats "ExtraSection" feature.

We're not actually using the Fedora Tomcat logs because we have front end Tomcat app. that sits out in front of Fedora and sends requests back to the repository, but the method is applicable I think.

Basically, we create a regular expression condition in an AWStats "ExtraSection" that parses a Tomcat request out in our front end application. We parse out the particular method call and the object identifier, and some front end specific information and create separate reports by object type (i.e. TEI texts, images, PDFs). You could probably even parse out BDEF identifiers in this way as well for more fine grained analysis.

This gets us:
which objects are accessed the most within a given class (text, image, etc.)
which overall objects are most accessed
how many time individual objects have been accessed.

We'll do the same for collections eventually.

With some additional processing over these logs (in Perl), we've generated RSS feeds that present overall lists of most accessed objects for the day – although, this is still a bit experimental.

It would be interesting to experiment with combining

Tomcat/IIS/Apache log files with the Fedora RI (since Dublin Core and a lot of other useful info is stored there) to see what kind of reports one could generate.

The live statistics for the Tufts Fedora instance may be useful.

Scroll all the way to the bottom to find the compiled text, image, and PDF results. We're in the process of implementing collection objects, but these aren't represented in the logs yet.

Here's one of our test RSS feeds for most accessed texts. This is still a work in progress and won't update daily yet at this point. I'm working on some enhancements before I make it live in the sense that it's updated daily. For now, I manually fire off the process as I debug.

http://dca.tufts.edu/rss/textfeed.xml

Some notes:

1. URL: the URL I'm matching against is from the app. that sits in front of Fedora; it looks something like this: http://dl.tufts.edu/view_text.jsp?urn=tufts:central:dca:UA069:UA069.005.DO.00001
This URL varies a bit depending on the type of object that's being access, so I've created separate ExtraSections in the AWStats config file for these. The text section looks like this:

1. a. XML text URN stats
ExtraSectionName1="Compiled XML Text URN hits"
ExtraSectionCodeFilter1=""
ExtraSectionCondition1="URL,\/view_text\.jsp"
#ExtraSectionCondition1="URL,\/dl\/view_text\.jsp|URL,\/dl\/view_image\.jsp|URL,\/dl\/view_pdf\.jsp"
ExtraSectionFirstColumnTitle1="URN"
ExtraSectionFirstColumnValues1="QUERY_STRING,urn=(&+)"
ExtraSectionFirstColumnFormat1="%s"
ExtraSectionStatTypes1=H
ExtraSectionAddAverageRow1=0
ExtraSectionAddSumRow1=1
MaxNbOfExtra1=100
MinHitExtra1=1

Basically, I match on the type of object (indicated by the call to view_text in this case, which indicates our Fedora text BDEF, bdef:TuftsText, is being called). Then I grab the URN, which we map to the Fedora PID. I could collect calls to the object's metdata in the same way, but for the moment we're only interested in overall requests for the object that come through the DL front end app.

It would be simple to configure the ExtraSection to work against Fedora URLs themselves as opposed to a front end app. It would just be a matter of parsing the Fedora Tomcat URLs and figuring out what you want to match and collect:

For example, here's the Fedora call that corresponds to our front end app URL:
/fedora/get/tufts:UA069.005.DO.00001/bdef:TuftsText/getChunk?chunkID=d.1925.su.Whilte

We could just grab the PID and the BDEF and come up with the same results.

2. We're using AWStats 6.4. The latest version of AWStats has some new features that specifically provide for daily log reports in addition to cumulative monthly reports. I haven't had the time to try this version yet, but I think it would make the collection of daily Fedora object reports even easier, especially if you wanted daily RSS feeds.

3. RSS: the RSS feeds are an addition process which I do in Perl. I use the AWStats daily log file and parse out the top requested objects (text objects in the above example) collect the URNs, request the titles/labels from the Fedora metadata and then put the whole thing together in XML. It's a work in progress at the moment. Again, I think the daily reporting features in latest version of AWStats would make this process simpler.