From Ingest To Access: A Day In The Life Of A HathiTrust Object

Notes:

1. Bibliographic data must be loaded into Aleph before content is ingested. This data has a number of requirements:
    a. Records as complete as possible
    b. One bibliographic record per item (multi-volume works should have the same record repeated for each item). Each record should contain a single 955 field
    c. Local system number in 001
    d. OCLC number in an 035 field with appropriate identifying prefix (OcoLC, ocm, ocn, etc.)
    e. Barcode in 955 l b; any alphabetic characters should be lowercase.
    f. Item description (enumeration / chronology) in 955 l v

    As bib data is ingested, a process identifies records that already exist in the repository based on OCLC number. If a bib record already exists, new holdings and item records (for the ingested items) are added to the existing record.

2. Bibliographic Content Store (Aleph)
    a. After bibliographic ingest has taken place, when content matching a bibliographic records also enters the repository, a process is triggered that adds the volume id and a timestamp to a field in the bibliographic record and starts an automated rights determination process based on bibliographic metadata (publisher date, publisher, etc.).

3. Rights Determination
    a. When the rights determination process is complete, a list of the volume ids and other rights information is inserted into the rights database.

4. MySQL Database Server
    a. This is where the rights, shadow rights, and Geo IP databases are located.

5. Rights Database
    a. For a full description of the database, see http://www.hathitrust.org/rights_database
    b. The results of automated and manual copyright review process are stored here.

    c. There are multiple entries in the Rights Database for a single item if the rights information of that item changes (i.e was determined automatically to be in copyright, but on review is determined to be in the public domain).

6. Shadow Rights Database
    a. This database contains rights information and the most recent time the information was created or changed for every volume in the repository. A timestamp is maintained that allows a query for the ids whose rights have changed since or which entered the repository for the first time since the last time indexing occurred. It is used to generate the list of items at any given time that can be placed in a separate queue to be indexed for large-scale search.

7. Geo IP database
    a. Viewing privileges for volumes that are public domain in the US are different if users are viewing them from inside or outside of the United States. This database contains mapping information that the PageTurner uses to determine user access.

8. Processes check for updated information in bib records and in the rights database daily to be sure all rights information is current.

9. Bibliographic Search Solr Index
    a. The temporary HathiTrust catalog is built on the VuFind platform, which uses Solr search. The Solr search index is rebuilt daily from the primary bibliographic data store to capture changes that have been made.

10. Queries are made to this index whenever a user query is entered.

11. Catalog User Interface
    a. The temporary catalog provides bibliographic search and faceted browsing for all volumes in HathiTrust.

12. Content Ingest
    a. Although this appears as number 12 on the list, the ingest of content is the trigger for all processes following bibliographic ingest. The bibliographic data is loaded and rights determination, indexing, availability, etc.. follow after the corresponding volumes have been ingested.
    b. The only master formats currently in the repository are ITU G4 TIFF and JPEG2000.

13. HathiTrust has an array of servers devoted to high volume backend processing. These servers currently handle content ingest (with a capacity to ingest up to 500,000 volumes a month) and validation, and large scale search indexing.

14. GROOVE (Google Return Object-Oriented Validation Environment) is a custom-built ingest mechanism that handles inbound validation, METS file creation, the creation of a quality review sample, handle assignment, and Zip file creation for every volume that enters HathiTrust. The following validation is done:
    a. Luhn validation on barcodes
    b. Fixity check on JPEG2000, TIFF, UTF-8 using MD5
    c. Well-formedness and embedded metadata check on JPEG2000, TIFF, UTF-8 using JHOVE

    And the METS file that is created is composed of
    a. metsHdr with an ID, creator of METS document, and creation date
    b. dmdSec with marcxml
    c. dmdSec with a reference to ILS record
    d. amdSec containing one techMD with PREMIS metadata
    e. fileSec with four fileGrps (zip archive, images, OCR, and coordinate OCR)
    f. Physical structMap connecting files with metadata (pg. numbers or features)

15. Quality review on deposited content is the responsibility of the depositing institution (see http://www.hathitrust.org/quality for more information about quality). The University of Michigan has been doing longitudinal analysis of quality on Google-scanned volumes since 2007 as part of its partnership with Google. Results of this analysis are fed back to Google, resulting in quality improvements across volumes digitized from all Google-partner libraries. Michigan's process is represented here. When volumes are ingested, a sample of 20 consecutive pages, randomly selected within each volume, is set aside for quality review. Manual QR is performed on approximately 1% of these samples, reviewing for problems such as blur, cleaning, warp, crop, obscuring of text or images, and colorization issues, as well as errors relating to thresholding (thick or broken).

16. Content Data Store
    a. After passing all of the steps involved in GROOVE, the following elements enter the repository:
        i. A Zip file containing
            1. Page image files
            2. OCR files
            3. Coordinate OCR files

4. A Google METS document
   ii. A HathiTrust METS file
b. This content is stored in a pairtree directory structure (http://www.cdlib.org/inside/diglib/pairtree/pairtreespec.html) on Isilon storage.
c. Features of the storage include:
   i. Built for disaster recovery. High redundancy.
   ii. Divided into nodes (CPUs + storage)
   iii. Data is distributed across all nodes
   iv. N+3 parity protection = 3 nodes can completely fail and all data will still be available
   v. Sync IQ replication software
   vi. MediaScan scans blocks in the system for bad disk sectors. If it finds one, it uses parity information to rebuild the necessary data and rewrite a block somewhere else on the drive. Content is migrated to a block in a new node and re-balanced across the system, incorporating the new node.
   vii. OneFS 6.0 operating system release will allow checks of system data and metadata via associated checksums.

17. SLIP (Solr Large-scale Indexing Processer)
    a. SLIP creates documents for indexing. The full process is that SLIP
       i. Receives a list of ids to be indexed from the Shadow Rights Database.
       ii. Queries the repository, opens the zip file, unzips the OCR
       iii. Queries VuFind Solr index for Solr metadata fields
       iv. Queries Rights Database for rights information
    b. After building a document, SLIP sends it to one of the Large scale search Solr servers in round robin fashion for indexing. There are N number of processes for creating documents in each SLIP instance on the ingest servers.

18. Large-scale Search Servers
    a. There are four Solr instances on each large-scale search server that handle two index shards. Two Building Solr instances build one shard each. Once a day, snapshots of the shards are mounted and served by the Serving Solr Instances. The Serving Solr instances receive queries from the LSS application (22) and send them to the shards on all of the other machines. The Serving Solr instances merge query results from the shards and send the results back to the application.

19. Web Servers

a. Software for the PageTurner, Collection Builder, and Large-scale Search applications is located on web servers at each repository location (University of Michigan and Indiana University).

20. Large-scale Search Application
    a. This application resides on the web servers at each repository location (UM and IU). It sends user queries to the Solr Serving instances and receives and displays results.

21. Collection Builder
    a. The Collection Builder application also resides on the repository web servers. It allows users to save items from the repository to public or private collections, and to perform full-text search inside of those collections.
    b. It works by saving limited metadata for items that have been saved to a collection (title, publisher, date) to a database table with the collection to which it belongs.
    c. When new items are added, the volume OCR for those volumes is retrieved from the repository and indexed (or reindexed) along with other volumes from that collection.

22. Collection Builder Solr Index
    a. The index of all volumes that have been saved to a collection in Collection Builder.

23. PageTurner
    a. When a volume in HathiTrust is accessed for viewing, the PageTurner
        i. Retrieves bibliographic information for that volume from the bibliographic data store
        ii. Retrieves source and attribute information from the rights database to determine access capabilities (search-only or full-text)
        iii. Retrieves the corresponding object's METS file, and the page image or OCR text that is requested by the user. Page images and OCR are extracted from the repository Zip file and page images are transformed on the fly into access-quality images.

24. Accessible Interface
    a. HathiTrust has configured an accessible interface to its content. Please see the diagram for more information.

25. Computational Research

a. HathiTrust has defined three different methods for allowing computational research on repository content
    i. Data distribution (for public domain volumes – samples are currently available on the HathiTrust website at http://www.hathitrust.org/datasets
    ii. A protocol-based method (e.g. SEASR) that will allow researchers to run routines on public domain content inside the repository and receive results.
    iii. A research center that will provide researchers the capability to do intensive processing across the entire body of repository materials.