Search

Browse (/directory/)   **Enterprise (/directory/enterprise)**   **Blog (/blog/)**   **Log In (/auth/)**   or   **Join (**

Jobs (/jobs?source=header)   **Deals (//deals.sourceforge.net/?**
utm_source=sourceforge&utm_medium=navbar&utm_campaign=homepage)

SOLUTION CENTERS   Go Parallel (http://goparallel.sourceforge.net/)   P

Help (/support)

Home (/) / Browse (/directory) / Projects (/p/) / The Lemur Project (/p/lemur/) / Wiki

The Lemur Project (/p/lemur/)

Brought to you by: david_fisher (/u/userid-1463830/), jamanjeval (/u/jamanjeval/), sm-harding (/u/sm-harding/)

Summary (/projects/lemur/) | Files (/projects/lemur/files/) | Reviews (/projects/lemur/reviews) | Support (/projects/lemur/support) |

Wiki (/p/lemur/wiki/) | Discussion (/p/lemur/discussion/) | Tickets ▾ (/p/lemur/_list/tickets) | Code (/p/lemur/code/) | Galago (/p/lemur/galago/) |

Cvs (/p/lemur/cvs/) | Mailing Lists (/p/lemur/mailman/)

Search Wiki

Wiki Home
(/p/lemur/wiki/)

Browse Pages
(/p/lemur/wiki/browse_

Browse Labels
(/p/lemur/wiki/browse_

Formatting Help
(/p/lemur/wiki/markdov

Indri Repository Structure   (history)   (feed)   (../search)

Authors: (/u/userid-1463830/)

Overview

An Indri Repository is a collection of a set of files with a specified format that contains all the relevant information regarding a collection. The collection contains information about the indexed documents, any fields or metadata, the inverted indexes for the collection and other necessary items. It is important to note that when you request to open an Indri index, there is no one specific file that you should give, but rather, the root of the collection folder structure should be used.

Technical Details

While building an Indri index, the indexer will build indexes in memory before writing them out to disk. This increases the speed of indexing, and allows the indexer to flush only when ready (or necessary) from a separate thread. As collections are indexed, the indexer will keep multiple indexes in memory which act as one repository.

The indexer will automatically merge in-memory indexes when the soft-limit for memory is reached. Indri will typically also merge indexes after a few very big documents or a lot of very small documents. For example, if you are using a gigabyte of memory, I would guess that Indri would write to disk after about 100,000 documents.

When merging two small indexes, Indri always chooses to merge the most recent index with the one before it. In many cases, though, Indri will choose to merge many indexes together at once (as many as 50). The last index is always included.

The detailed explanation of the Indri repository structure and index build can be found in the paper Dynamic Collections in Indri (http://ciir.cs.umass.edu/pubfiles/ir-426.pdf) (PDF format) by Trevor Strohman.

For details of the index building and merging operations, see Low Latency Index Maintenance in Indri (http://ciir.cs.umass.edu/pubfiles/ir-503.pdf) (PDF format), also by Trevor Strohman.

Disk Structure

On disk, an Indri collection is made up of several files:

- Frequent Vocabulary Files:
    - for any term that appears more than 1000 times in a corpus
    - File: "frequentID" - a !BulkTree structure (essentially a B-Tree) mapping from termID to a term string. The value entries also store the start offset in the inverted list file and the length of the entry in the inverted list file.
    - File: "frequentString" - a !BulkTree structure mapping from term string to a termID. The value entries also store the start offset in the inverted list file and the length of the entry in the inverted list file.
    - File: "frequentTerms" - a list (not a tree) of tuples having <termID, term="" string=""> for each pair - used only at index merge time while building a collection.

- Infrequent Vocabulary Files:
    - File: "infrequentID" - a !BulkTree structure mapping from termID to a term string. The value entries also store the start offset in the inverted list file and the length of the entry in the inverted list file.
    - File: "infrequentString" - a !BulkTree structure mapping from term string to a termID. The value entries also store the start offset in the inverted list file and the length of the entry in the inverted list file.

- Inverted Lists
    - File: "invertedFile" - the inverted lists for all terms in the collection. This file consists of (for each term):
        - corpus statistics such as the doc frequency and corpus frequency and statistics for each field < doc frequency, corpus frequency >
        - the maxDocumentLength and minDocumentLength that the term occurs in
        - the actual term string
        - top document ID list (top 1% of the documents my document frequency of the term) in
        - the actual inverted list (in RVL Compressed format) consisting of:
            - docID (delta-encoded from previous docID)
            - size of position data
            - the actual position data - 1 integer per position in this document (also delta encoded from previous position)

- Field Information File - The inverted lists file for all fields in the collection. This file consists of (for each field):
  - like the regular inverted lists, RVL Compression used and for each entry:
    - the docID (delta encoded from last doc ID)
    - number of extents in the document and for each extent:
      - extent begin (delta-encoded from last begin)
      - extent end (delta-encoded from last end)
      - extent ordinal
      - numeric value (if applicable)

Related

Wiki: Home (/p/lemur/wiki/Home/)
Wiki: Indri (/p/lemur/wiki/Indri/)
Wiki: Technical Details (/p/lemur/wiki/Technical%20Details/)