Robert Isele
mail@robertisele.com

Christian Bizer
chris@bizer.de

Jürgen Umbrich
juergen.umbrich@deri.org

Andreas Harth
harth@kit.edu

Freie Universität Berlin

LDSpider - An open-source Crawling Framework for the Web of Linked Data

LDSpider is an extensible Linked Data crawling framework, which enables client applications to traverse and to consume the Web of Linked Data. LDSpider provides an easy-to-use command line interface as well as a Java API which allows applications to configure and control the details of the crawling process.

Features

- LDSpider can process a variety of Web data formats including RDF/XML, Turtle, Notation 3, RDFa and many microformats.
- Crawled data can be stored together with provenance meta-information either in a file or via SPARQL/Update in an RDF store.
- LDSpider offers different crawling strategies, such as breadth-first traversal and load-balancing, for following RDF links between data items.
- Besides of being usable as a command line application, LDSpider also offers a simple API which allows applications to configure and control the details of the crawling process.
- The framework is delivered as a small and compact jar with a minimum of external dependencies.
- LDSpider is high-performing by employing a multi-threaded architecture.
- Can be used to collect small to medium-sized datasets up to tens of millions of triples.

Architecture

Scheduler

Traverses the graph starting from the seed URIs as specified by the crawling strategy.

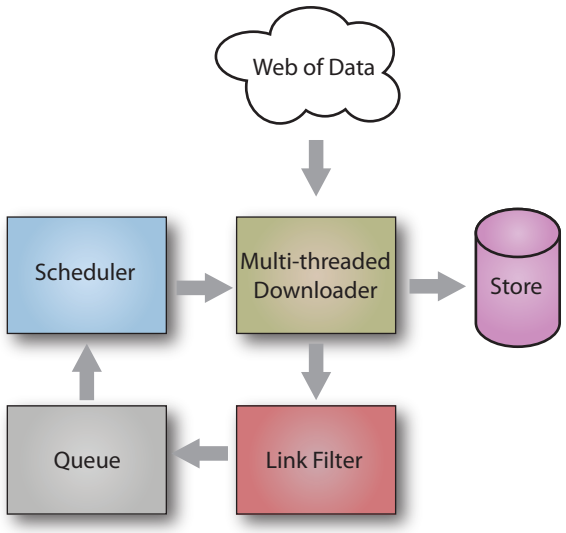
Breadth-First Strategy

- Fetches all URIs extracted in the previous round, before advancing to the next round.
- This strategy can be used in situations where only a limited graph around the seed URIs should be retrieved.

Load-Balancing Strategy

- Tries to fetch the specified number of URIs as quickly as possible while adhering to a minimum and maximum delay between two successive requests to the same pay-level domain.
- Useful in situations where the fetched documents should be distributed between domains without overloading a specific server.

Both strategies can be configured to stay on the domains of the seed URIs.



Downloader

Receives the document and tries to extract RDF data from it.

- Fetches URIs in parallel employing multiple threads
- Supports Any23 server to handle other types of documents including Turtle, Notation 3, RDFa and many microformats.

Link Filter

Extracts links from the document which should be fetched in the next round

- Can be configured to follow only ABox and/or TBox links.
- Allows to restrict the crawling to a specific domain.

Output

The extracted statements can be written to different outputs.

Triple Store Output

- Can write to every store supporting SPARQL/Update
- The provenance can be included using a Named Graph layout

File Output

- N-Quads and RDF/XML
- Optionally includes the provenance of the extracted document

Usage Examples

Crawling FOAF profiles

Crawling of interlinked FOAF profiles starting with single seed profile.

round	1	2	3	4	5
profiles	1	10	101	507	6730

Crawling Twitter profiles

Crawling of interlinked Twitter profiles, which expose structured data using RDFa, starting with a single seed profile.

round	1	2	3
profiles	1	38	1160

Further information and download: **http://code.google.com/p/ldspider/**