HDT

Queryable compression format for Linked Data

Wouter Beek, VU University Amsterdam

Javier D. Fernández, Vienna University of Economics and Business, Austria

Ruben Verborgh, Ghent University – imec, Belgium

ISWC 2017

21TH SEPTEMBER 2017

Practical tips

- WIFI
  - SSID: wu-conference
  - Username: wu0041
  - Password: SWSA2017
- Get slides: https://rdfhdt.github.io/ISWC2017/
- Please write/follow the minutes at: https://board.net/p/HDT-ISWC17
- Twitter: @rdfhdt

General agenda

- **Session I** (09:00 - 9:40) Welcome and HDT foundations
- **Session II** (9:40 - 10:10) Practical uses I: Linked Data Fragments

- **Session III** (10:10 - 10:30) Practical uses II: LOD Laundromat

10:30 – 11:00

- **Session IV** (11:00 - 11:45) Presentations - Different perspectives on the topic

  - "Practical use of HDT within **data.world**". Bryon Jacob and Jonathan Ortiz, data.world
  - "HDT as the backend engine for **Query Answering** over the Web of Data". Dennis Diefenbach, Laboratoire Hubert Curien, Saint Etienne, France.
  - "Counting to K: **Top k Shortest Path** Queries with HDT". Vadim Savenkov, Vienna University of Economics and Business, Austria.

- **Session V** (11:45 - 12:30) Discussion: concrete next steps and closing remarks

img: zurb.com

Knowledge of HDT

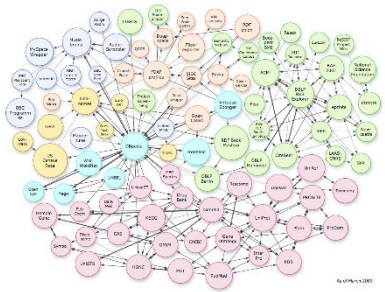
0. Zero knowledge
1. I have just heard of HDT, I know some of the potential application scenarios
2. I know the basic foundations of HDT and I gave it a try
3. I often manage HDT datasets and tools

img: Nick Youngson

Motivation. Origins

- ‘Simple’ task in 2009 (by Claudio Gutiérrez )
  - Let’s inspect what people are publishing in RDF
    - Find RDF datasets
    - Download them
    - Do some (simple) queries to inspect the content

2009



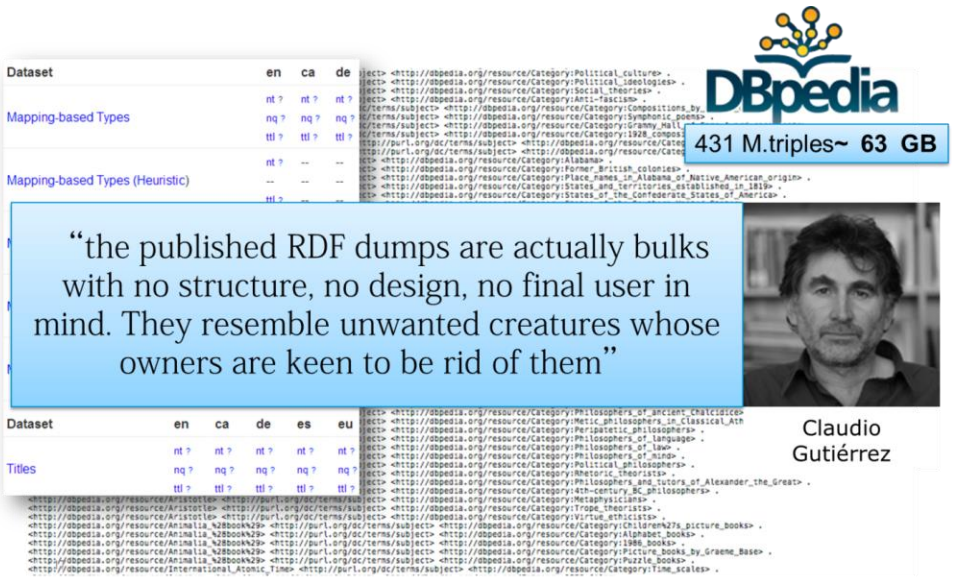
- Problems?
  - Discover datasets
  - Hugh resources to download (large) datasets
  - Messiness of the data
  - Hugh resources to index (large) datasets locally
  - Hugh resources to query (large) datasets locally and to serve them online

Is it much better now in 2017 ??

img: Beth Scupham

The problem is in the roots (Big tree = big roots)

- Publication, Exchange and Consumption of **large RDF datasets**
  - Most RDF formats (N3, XML, Turtle) are text serializations, designed for human readability (not for machines)
    - Verbose = High costs to write/exchange/parse
    - A basic offline search = (decompress)+ index the file + search



Header Dictionary Triples (HDT)

VU UNIVERSITY AMSTERDAM

WU WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS

imec

> HDT compresses big RDF datasets while maintaining search operations

"HDT might not be the most compressed format or the fastest RDF store, but it should be the most efficient way to publish, exchange and consume RDF datasets.
Like a duck, it may not walk, swim and fly as fast as other animals … but he does everything!"

Motivation

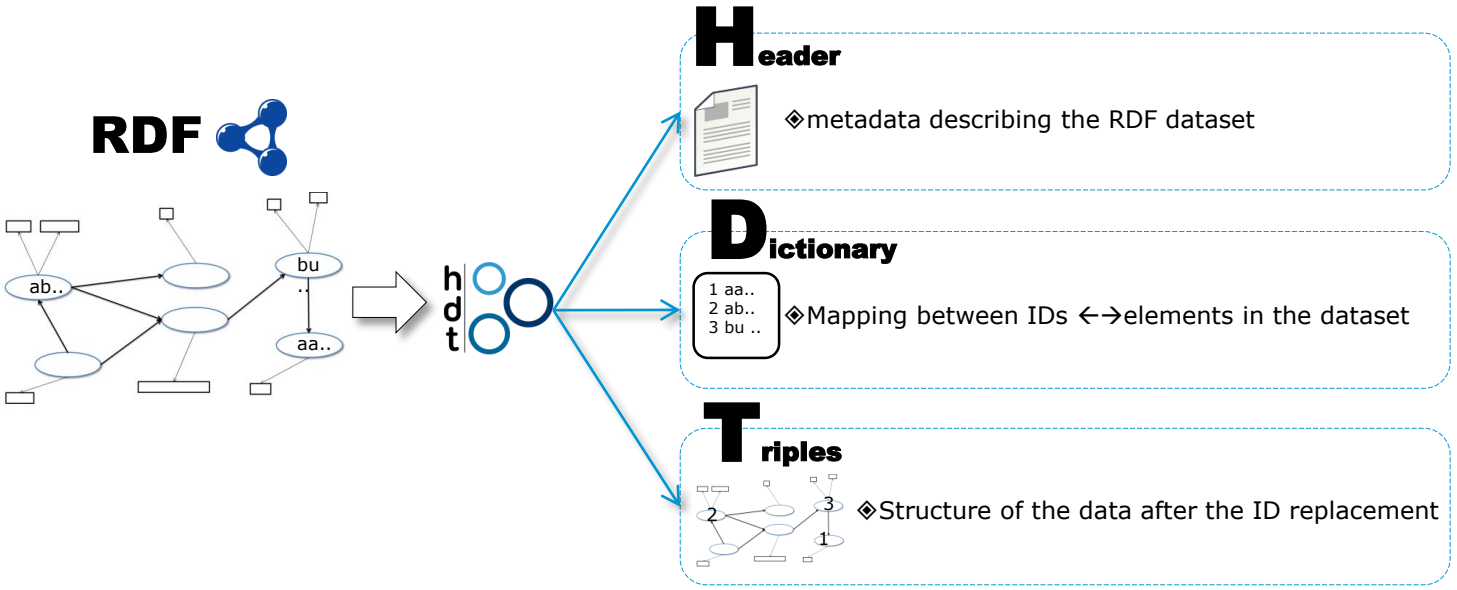
- Lightweight Binary RDF (HDT)
  - Highly compact serialization of RDF
  - Allows fast RDF retrieval in compressed space (without prior decompression)
    - Includes internal indexes to solve basic queries with small (3%) memory footprint.
    - Very fast on basic queries (triple patterns), x 1.5 faster than Virtuoso, RDF3x.
    - Complex queries (joins) on the same scale of current solutions (Virtuoso, RDF3x).

rdfhdt.org

Application scenarios

- Scalable storage
  - Store and serve thousands of (large) datasets (e.g. LOD Laundromat)
- Better consumer-centric publication
  - Compress and share ready-to-consume RDF datasets
  - Consumption with limited resources
    - smartphones, standard laptops
- Fast –low cost- SPARQL Query Engine
  - Via Linked Data Fragments (shown in next session)
  - Via HDT-Jena

HDT (Header-Dictionary-Triples) Overview



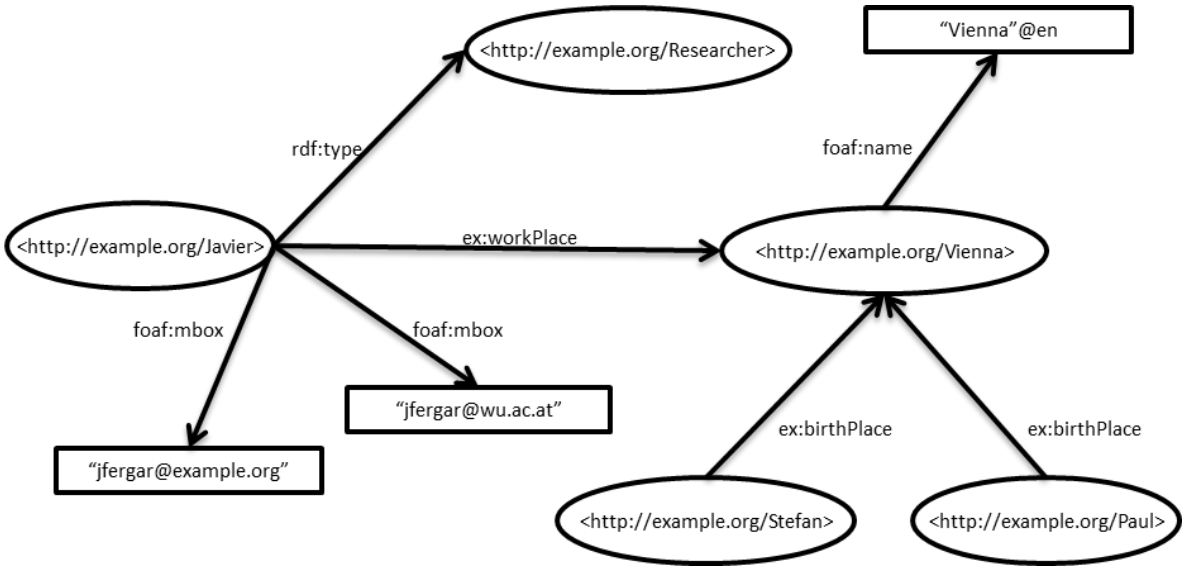
Header

- Publication Metadata:
  - Publisher, Public Endpoint, …
- Statistical Metadata:
  - Number of triples, subjects, entities, histograms…
- Format Metadata:
  - Description of the data structure, e.g. Triples Order.
- Additional Metadata:
  - Domain-specific.
  - … In RDF

Header

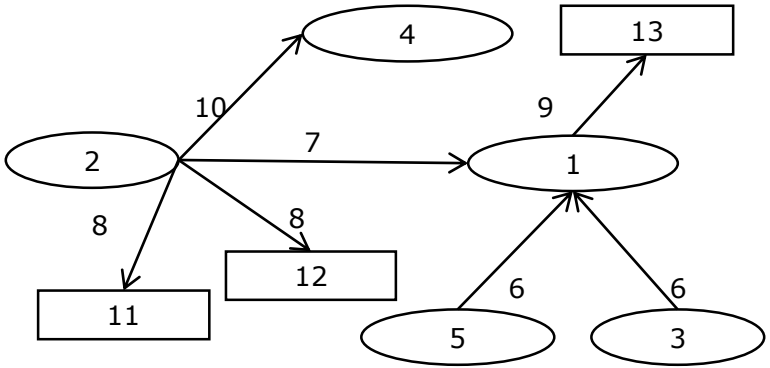
	Subject	Predicate	Object
1	<http://example.org>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://purl.org/HDT/hdt#Dataset>
2	<http://example.org>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://rdfs.org/ns/void#Dataset>
3	<http://example.org>	<http://rdfs.org/ns/void#triples>	"4789981"
4	<http://example.org>	<http://rdfs.org/ns/void#properties>	"212"
5	<http://example.org>	<http://rdfs.org/ns/void#distinctSubjects>	"505761"
6	<http://example.org>	<http://rdfs.org/ns/void#distinctObjects>	"1717095"
7	<http://example.org>	<http://purl.org/HDT/hdt#statisticalInformation>	_:statistics
8	<http://example.org>	<http://purl.org/HDT/hdt#publicationInformation>	_:publicationInformation
9	<http://example.org>	<http://purl.org/HDT/hdt#formatInformation>	_:format
10	_:format	<http://purl.org/HDT/hdt#dictionary>	_:dictionary
11	_:format	<http://purl.org/HDT/hdt#triples>	_:triples
12	_:dictionary	<http://purl.org/dc/terms/format>	<http://purl.org/HDT/hdt#dictionaryFour>
13	_:dictionary	<http://purl.org/HDT/hdt#dictionarynumSharedSubjectObject>	"383152"
14	_:dictionary	<http://purl.org/HDT/hdt#dictionarymapping>	"1"
15	_:dictionary	<http://purl.org/HDT/hdt#dictionarysizeStrings>	"22120359"
16	_:dictionary	<http://purl.org/HDT/hdt#dictionaryblockSize>	"16"
17	_:triples	<http://purl.org/dc/terms/format>	<http://purl.org/HDT/hdt#triplesBitmap>
18	_:triples	<http://purl.org/HDT/hdt#triplesnumTriples>	"4789981"
19	_:triples	<http://purl.org/HDT/hdt#triplesOrder>	"SPO"
20	_:statistics	<http://purl.org/HDT/hdt#originalSize>	"688768256"
21	_:statistics	<http://purl.org/HDT/hdt#hdtSize>	"41646252"
22	_:publicationInformation	<http://purl.org/dc/terms/issued>	"2015-09-03T03:23:05+0200"

Dictionary+Triples partition



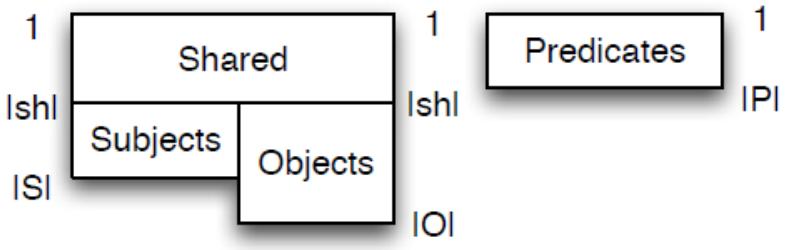
Dictionary+Triples partition

1	<http://example.org/Vienna>
2	<http://example.org/Javier>
3	<http://example.org/Paul>
4	<http://example.org/Researcher>
5	<http://example.org/Stefan>
6	ex:birthPlace
7	ex:workPlace
8	foaf:mbox
9	foaf:name
10	rdf:type
11	“jfergar@example.org”
12	“jfergar@wu.ac.at”
13	“Vienna”@en



Dictionary

- Mapping of strings to correlative IDs. {1..n}
- Lexicographically sorted, no duplicates.
- Prefix-Based compression in each section.
  - Efficient ID←→String operations



Dictionary

ID		
1	<http://example.org/Vienna>	SO
2	<http://example.org/Javier>	S
3	<http://example.org/Paul>	
4	<http://example.org/Stefan>	
2	<http://example.org/Researcher>	O
3	"jfergar@example.org"	
4	"jfergar@wu.ac.at"	
5	"Vienna"@en	
1	ex:birthPlace	P
2	ex:workPlace	
3	foaf:mbox	
4	foaf:name	
5	rdf:type	

Dictionary. Plain Front Coding (PFC)

- Prefix-Based compression used in each section
  1. Each string is encoded with two values
     - An integer representing the number of characters shared with the previous string
     - A sequence of characters representing the suffix that is not shared with the previous string

A
An
Ant
Antivirus
Antivirus Software
Best

➡ (0,a) (1,n) (2,t) (3,ivirus) (9, Software) (0,Best)

Dictionary. Plain Front Coding (PFC)

- Prefix-Based compression used in each section
  2. The vocabulary is split in buckets, each of them storing "*b*" strings
     - The first string of each bucket (*header*) is coded explicitly (i.e. full string)
     - The subsequent *b-1* strings (*internal strings*) are coded differentially

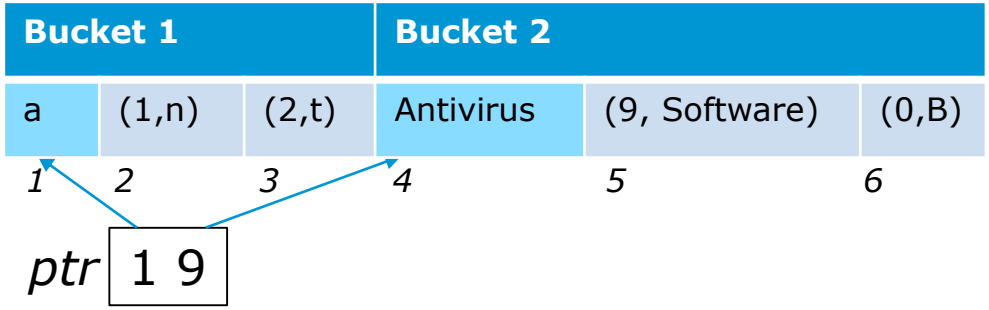
A
An
Ant
Antivirus
Antivirus Software
Best

Bucket 1			Bucket 2		
a	(1,n)	(2,t)	Antivirus	(9, Software)	(0,Best)
1	*2*	*3*	*4*	*5*	*6*

Dictionary. Plain Front Coding (PFC)

- Prefix-Based compression used in each section
  3. PFC is encoded with a byte sequence and an array of pointers (*ptr*) to denote the first byte of each bucket

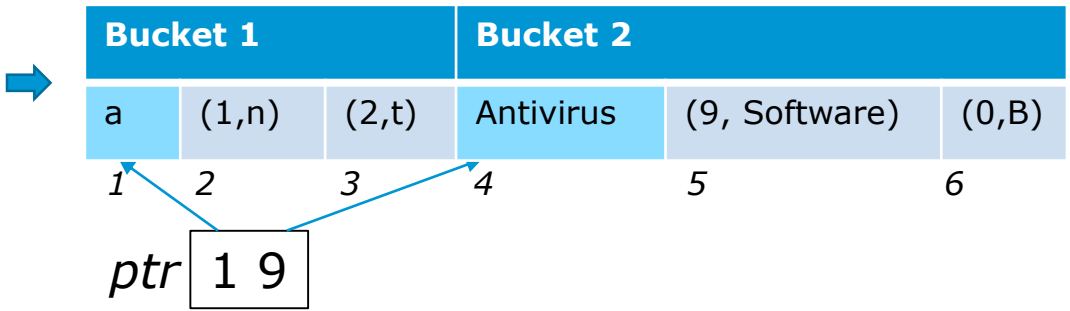
A
An
Ant
Antivirus
Antivirus Software
Best



Dictionary. Plain Front Coding (PFC)

- Prefix-Based compression used in each section
- Locate (string) performs a binary search in the headers + sequential decoding of internal strings
  - e.g. locate (Antivirus Software)=5
- Extract (id), finds the bucket id/b, and decodes until the given position
  - E.g. extract (5) = Antivirus Software

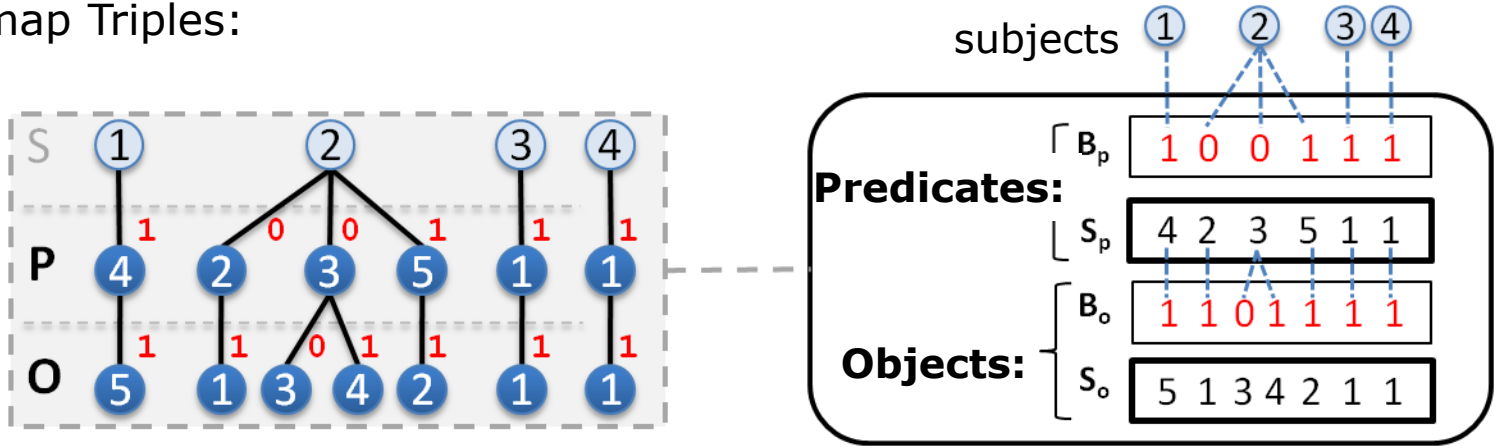
A
An
Ant
Antivirus
Antivirus Software
Best



More on Compressed Dictionaries: Martínez-Prieto, M. A., Brisaboa, N., Cánovas, R., Claude, F., & Navarro, G. (2016). **Practical compressed string dictionaries**. *Information Systems*, *56*, 73-108.

Bitmap Triples Encoding

- Bitmap Triples:



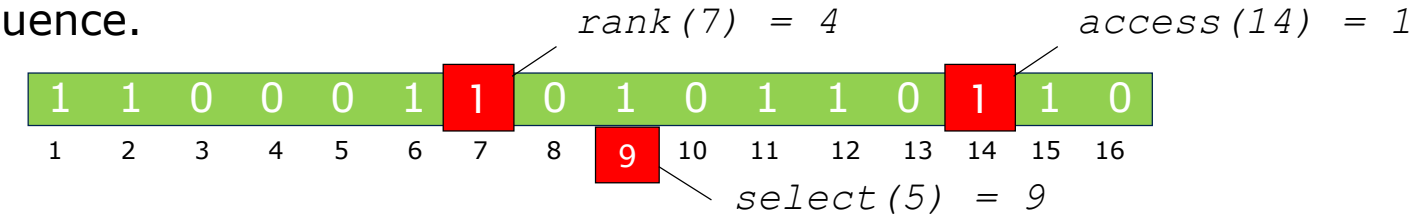
  - We index the bitsequences to provide a SPO index

Remember…. Succinct Data Structures

- Represent and index large volumes of data
- ~ Theoretical minimum space while serving efficient operations:
  - Mostly based on 3 operations:
    - Access
    - Rank
    - Select

Bit Sequence Coding

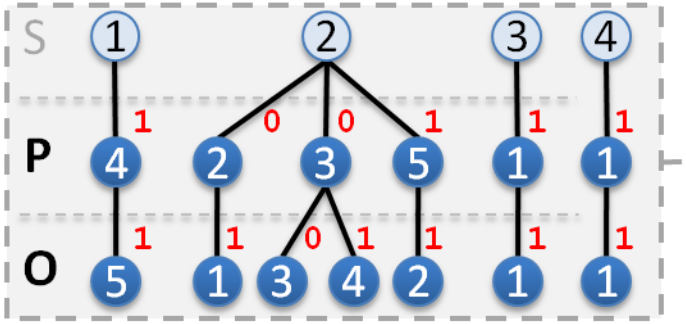
- Bitmap Sequence.

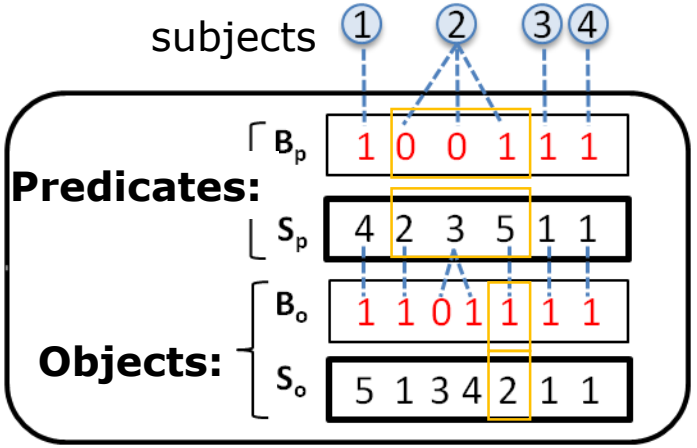


  - Operations in constant time
    - access(position) = Value.
    - rank(position) = "Number of ones, up to position".
    - select(i) = "Position where the one has i occurrences".
  - Implementation:
    - n + o(n) bits
    - Adjustable space overhead: In practice, 37,5 % overhead

Bitmap Triples Encoding

- Bitmap Triples:





- E.g. retrieve (2,5,?)
  - Find the position of the second '1'-bit in Bp (`select`)
  - Binary search on the list of predicates looking for 5
  - Note that such predicate 5 is in position 4 of Sp
  - Find the position of the four '1'-bit in Bo (`select`)

S P O

S P ?

S ? O

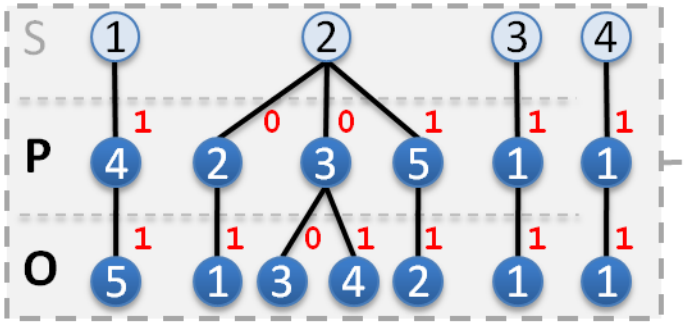
S ? ?

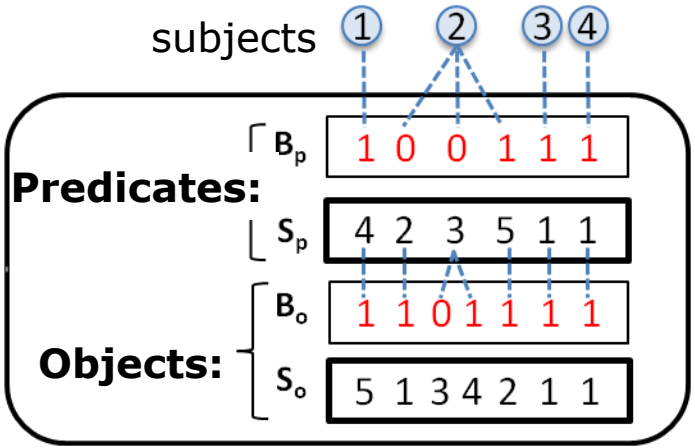
? ? ?

Additional Predicate Index

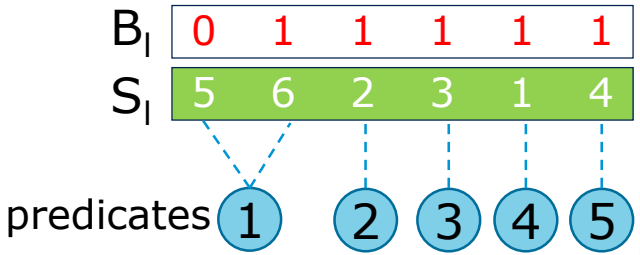
? P ?

- (Historically) ~~Wavelet Tree~~
  - Compact Sequence of Integers {1,|P|} with Operations in O(log |P|) time
- (Currently) Position list





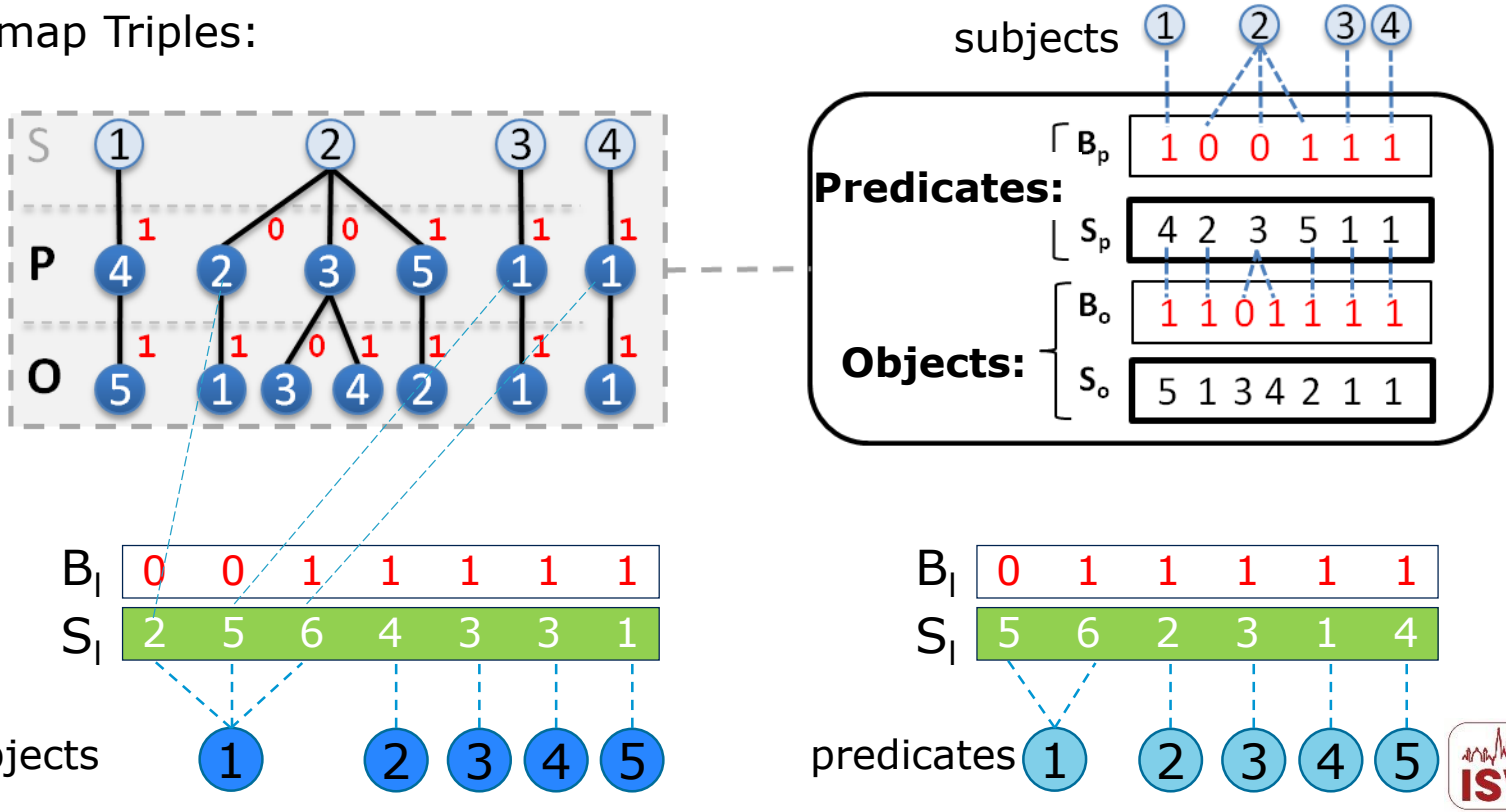
- E.g. retrieve (?,4,?)
  - Get the location of the list, B_I.select(3)+1= 4+1 =5
  - Retrieve the position: $S_I[5]=1$
  - *Get the associated subject, rank(1)=1st subject*
  - access the object as (1,4,?)



Additional Object Index

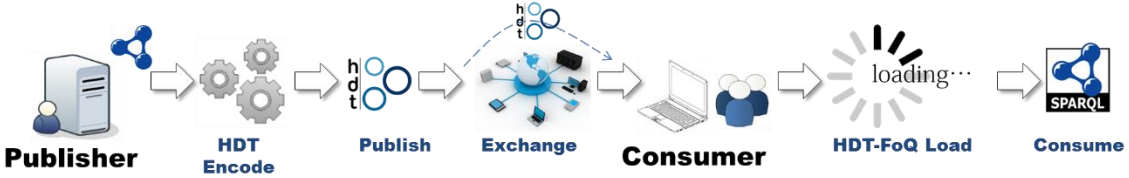
? ? O
? P O

- Bitmap Triples:



On-the-fly indexes: HDT-FoQ

- From the exchanged HDT to the functional HDT-FoQ:
  - Publish and Exchange HDT
  - At the consumer:



	Process	Type of Index	Patterns
1	index the bitsequences	Subject SPO	SPO, SP?, S??, S?O, ???
2	We index the position of each predicate (with a wavelet*)	Predicate PSO	?P?
3	We index the position of each object (just a position list)	Object OPS	?PO, ??O

HDT

Achievements and Challenges

VU UNIVERSITY AMSTERDAM

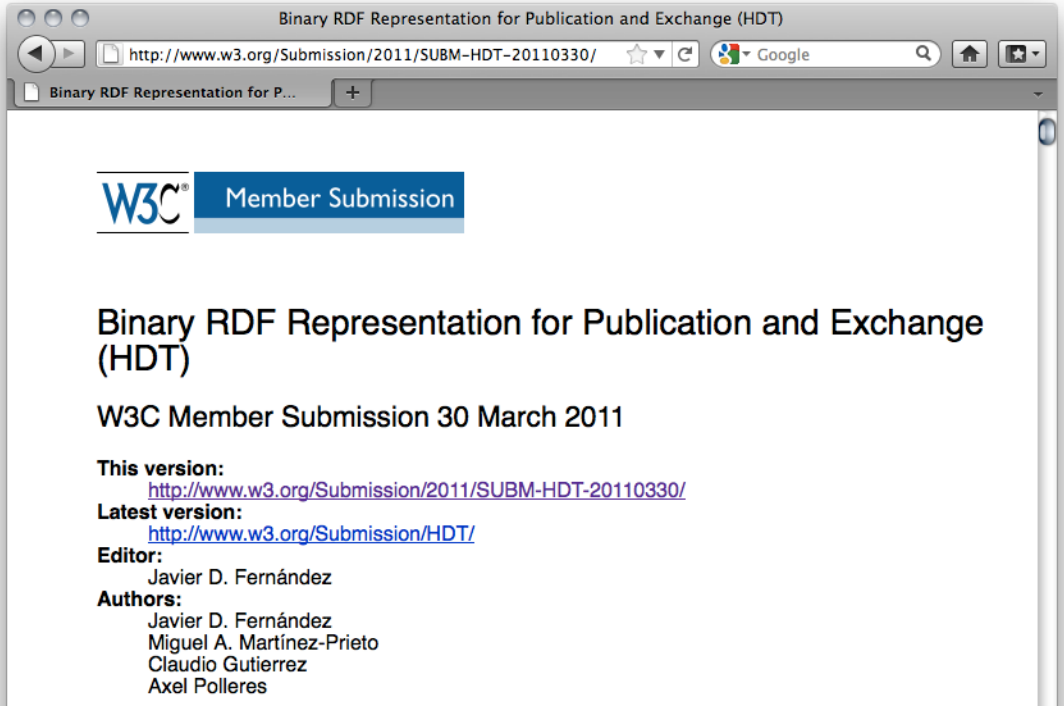
WU WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS

imec

Vienna · Austria ISWC 2017

HDT Acknowledged as W3C member submission:

- http://www.w3.org/Submission/2011/03/



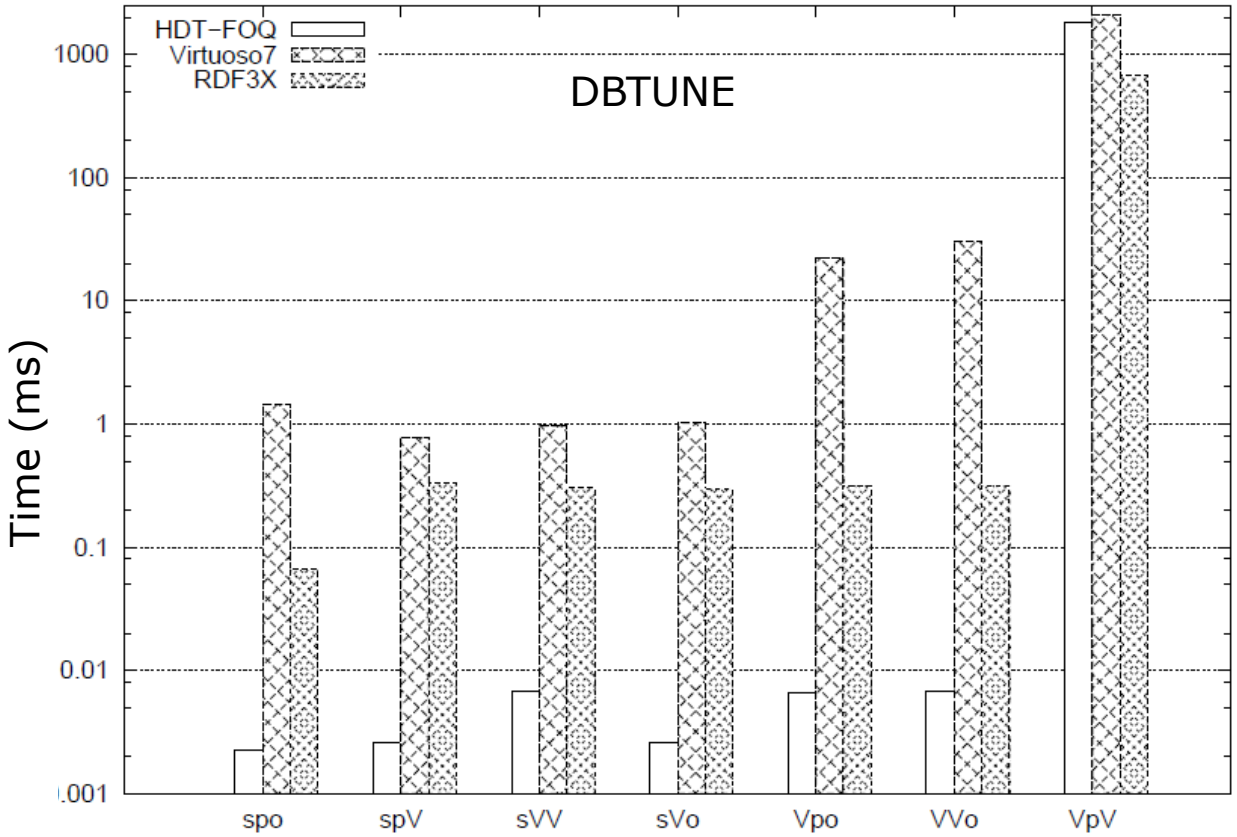
Some numbers on size

http://dataweb.infor.uva.es/projects/hdt-mr/

José M. Giménez-García, Javier D. Fernández, and Miguel A. Martínez-Prieto. **HDT-MR**: A Scalable Solution for RDF Compression with **HDT and MapReduce.** In Proc. of International Semantic Web Conference (ISWC),. 2015

						Size (GB)			
DATASET	TRIPLES	|SO|	|S|	|O|	|P|	NT	NT+lzo	HDT	HDT+gz
LinkedGeoData	0.27BN	41.5M	10.4M	80.3M	18.3K	38.5	4.4	6.4	1.9
DBPedia	0.43BN	22.0M	2.8M	86.9M	58.3K	61.6	8.6	6.4	2.7
Ike	0.51BN	114.5M	0	145.1K	10	100.3	4.9	4.8	0.6
Mashup	1.22BN	178.0M	13.2M	167.2M	76.6K	200.3	18.0	17.1	4.6
LUBM-1000	0.13BN	5.0M	16.7M	11.2M	18	18.0	1.3	0.7	0.2
LUBM-2000	0.27BN	10.0M	33.5M	22.3M	18	36.2	2.7	1.5	0.5
LUBM-3000	0.40BN	14.9M	50.2M	33.5M	18	54.4	4.0	2.3	0.8
LUBM-4000	0.53BN	19.9M	67.0M	44.7M	18	72.7	5.3	3.1	1.0
LUBM-5000	0.67BN	24.9M	83.7M	55.8M	18	90.9	6.6	3.9	1.3
LUBM-6000	0.80BN	29.9M	100.5M	67.0M	18	109.1	8.0	4.7	1.6
LUBM-7000	0.93BN	34.9M	117.2M	78.2M	18	127.3	9.3	5.5	1.9
LUBM-8000	1.07BN	39.8M	134.0M	89.3M	18	145.5	10.6	6.3	2.2
LUBM-12000	1.60BN	59.8M	200.9M	133.9M	18	218.8	15.9	9.6	2.9
LUBM-16000	2.14BN	79.7M	267.8M	178.6M	18	292.4	21.2	12.8	3.8
LUBM-20000	2.67BN	99.6M	334.8M	223.2M	18	366.0	26.6	16.3	5.5
LUBM-24000	3.74BN	119.5M	401.7M	267.8M	18	439.6	31.9	19.6	6.6
LUBM-28000	3.74BN	139.5M	468.7M	312.4M	18	513.2	37.2	22.9	7.7
LUBM-32000	4.27BN	159.4M	535.7M	357.1M	18	586.8	42.5	26.1	8.8
LUBM-36000	4.81BN	179.3M	602.7M	401.8M	18	660.5	47.8	30.0	9.4
LUBM-40000	5.32BN	198.4M	666.7M	444.5M	18	730.9	52.9	33.2	10.4

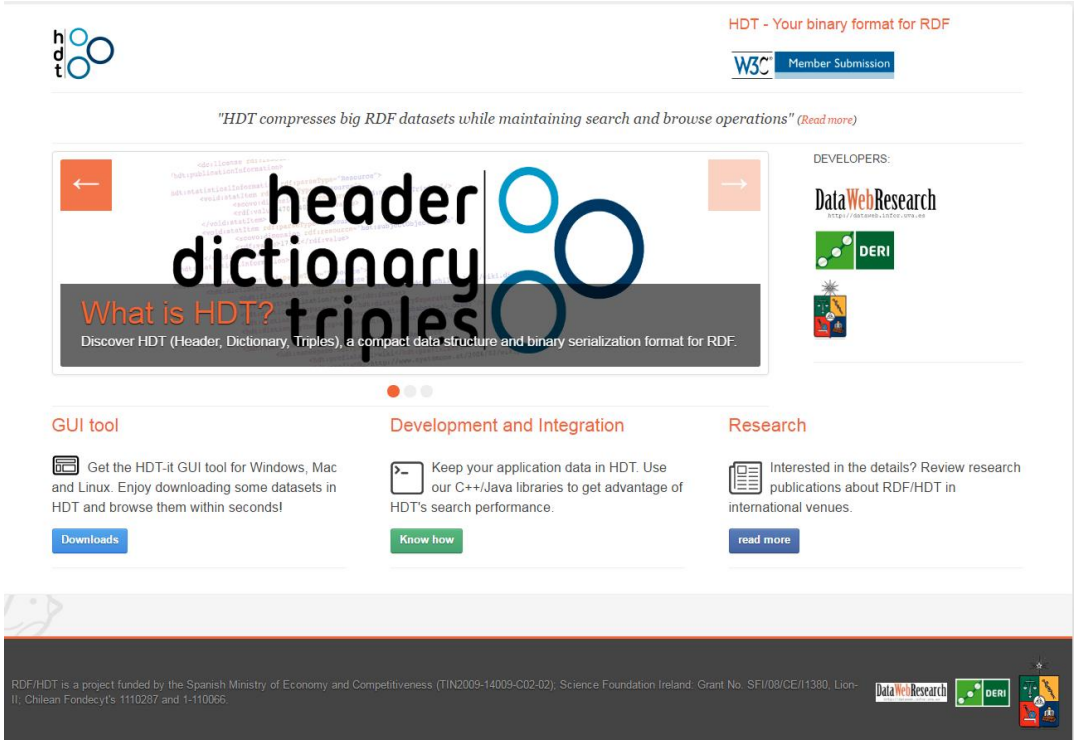
Evaluation. Querying (TP)



Results

- Data is ready to be consumed 10-15x faster.
  - HDT << any other RDF format || RDF engine
- Competitive query performance.
  - Very fast on triple patterns, x 1.5 faster (Virtuoso, RDF3x).
- Integration with Jena
  - Joins on the same scale of existing solutions (Virtuoso, RDF3x).

rdfhdt.org community



https://github.com/rdfhdt, C++ and Java tools

586 commits | 11 branches | 2 releases | 15 contributors

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

joernhees updated docker support & inlined ... Latest commit 9c807af 12 hours ago

hdt-it	Reduce string copying overhead	3 months ago
hdt-lib	Update README	8 days ago
libcds-v1.0.12	Compile on Windows 10 / VS2015. Have to install Windows 10 SDK and li...	7 months ago
.gitignore	Compile on Windows 10 / VS2015. Have to install Windows 10 SDK and li...	7 months ago
.travis.yml	Fix Travis build	3 months ago
Dockerfile	updated docker support & inlined	12 hours ago
README.md	add gitter chat	a month ago
maccompile.sh	Make sure it compiles in Mac.	7 months ago

369 commits | 6 branches | 3 releases | 13 contributors

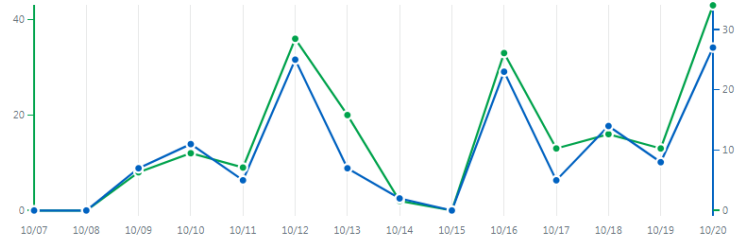
Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

joernhees dockerize java cli tools Latest commit 708f96b 7 hours ago

.settings	Added eclipse files to parent pom to run Maven targets directly	4 years ago
hdt-api	Merge pull request #28 from shawnsmith/parent-pom	7 months ago
hdt-fuseki	Jena 3.x code requires Java 8	a year ago
hdt-java-cli	Merge pull request #28 from shawnsmith/parent-pom	7 months ago
hdt-java-core	fix issue with HDT files remaining open in the system after calling c...	8 days ago

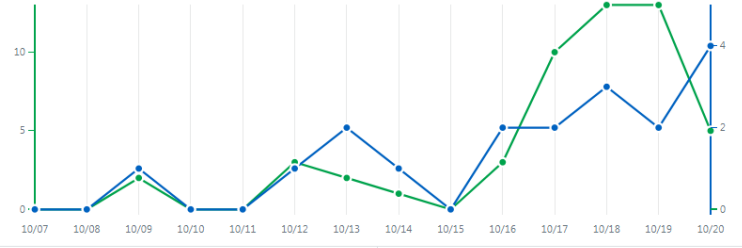
Only in the last two weeks…

Git clones — *HDT-cpp*



205 Clones | 110 Unique cloners

Git clones — *HDT-java*



52 Clones | 16 Unique cloners

Get more than 650K HDT datasets from LOD Laundromat…

LOD Wardrobe

This is where the cleaned data is stored. You can download both clean and dirty (i.e. the original) data. Each data document contains a meta-data description that includes all the stains that were detected.

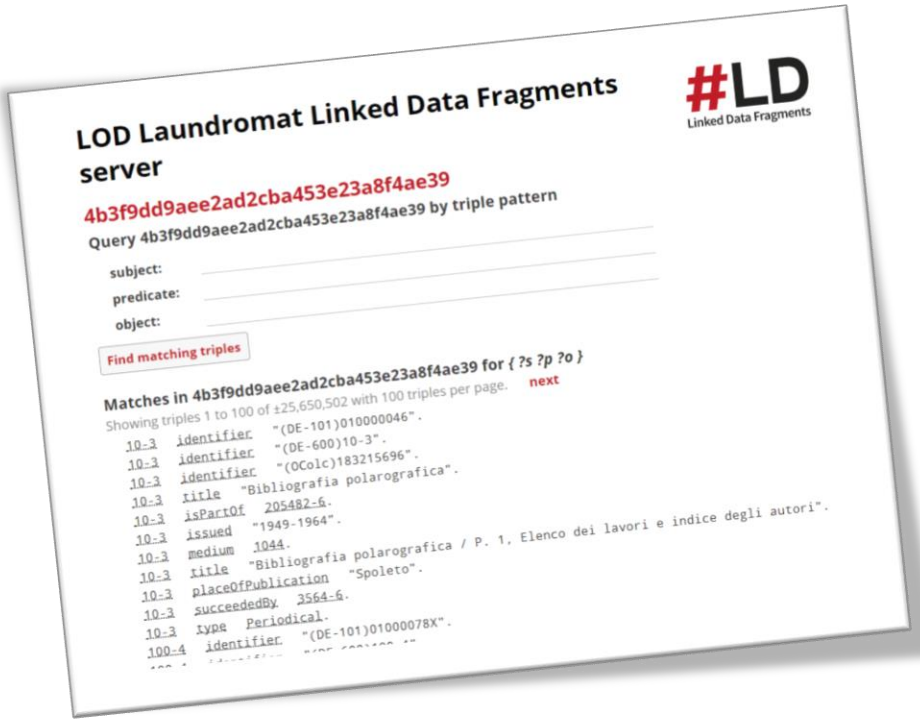
Show 10 entries

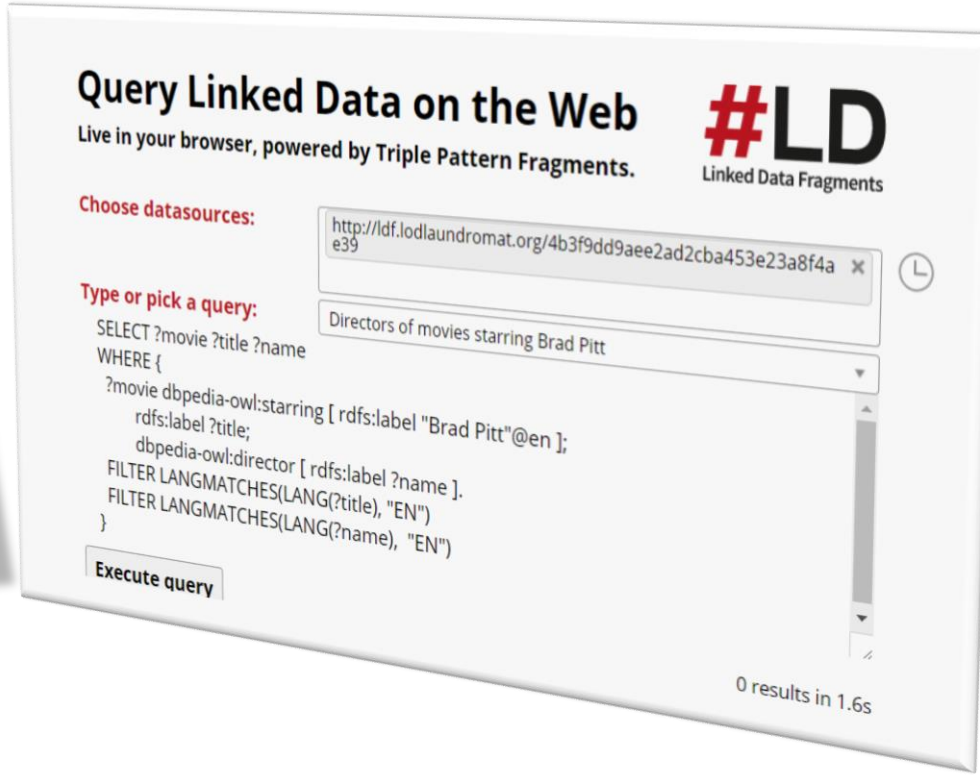
Url (filter)	Services	Download as	Triples (filter)
http://csarven.ca/	LD Browse · LD Query · Metadata	GZIP · HDT · Original Dirty File	54
http://www.example.com/iswc.rdf	LD Browse · LD Query · Metadata	GZIP · HDT · Original Dirty File	0
http://rdf.freebase.com/ns/m.012_t_	LD Browse · LD Query · Metadata	GZIP · HDT · Original Dirty File	16
http://datendienst.dnb.de/cgi-bin/mabit.pl?cmd=fetch&userID=opendata&pass=opendata&mabheft=ZDBTitel.ttl.gz	LD Browse · LD Query · Metadata	GZIP · HDT · Original Dirty File	25.650.502
http://prologmoo.com/downloads/mud.ttl	LD Browse · LD Query · Metadata	GZIP · HDT · Original Dirty File	4.707
http://franz.com/franzdownload/allegrograph/healthcare.nt.gz	LD Browse · LD Query · Metadata	GZIP · HDT · Original Dirty File	0
http://almere.pilod.nl/sparql?default-graph-uri=&query=CONSTRUCT+%0D%0A%7B%0D%0A++++%3Fa+geo%3Ageometry+%3Fc+.%0D%0A%7D%0D%0AWHERE%0D%0A%7B+%3Fa+geo%3Ageometry+%3Fc+.%7D%0D%0A&should-sponge=&format=text%2Fturtle&timeout=0&debug=on	LD Browse · LD Query · Metadata	GZIP · HDT · Original Dirty File	10.000
http://almere.pilod.nl/sparql?default-graph-uri=&query=construct%0D%0A%7Bgraph+%3Fg+%7B%3Fs+%3Fp+%3Fo%7D%7D%0D%0A%7Bgraph+%3Fg%0D%0A%7B%0D%0A%3Fs+%3Fp+%3Fo%0D%0A%7D%0D%0AFILTER+regex%28str%28%3Fg%29%2C%22%5Ehttp%3A%2F%2Fbag.kadaster.nl%2F%22%29%0D%0A%7D+LIMIT+10000&should-sponge=&format=text%2Fturtle&timeout=0&debug=on	LD Browse · LD Query · Metadata	GZIP · HDT · Original Dirty File	8.367
http://nxp.dydra.com/nxp/public-data.ttl	LD Browse · LD Query · Metadata	GZIP · HDT · Original Dirty File	1.630.760
https://www.dropbox.com/s/okzyej25j2aypkg/BEL2RDFexample.owl?dl=1	LD Browse · LD Query · Metadata	GZIP · HDT · Original Dirty File	33.150

Showing 1 to 10 of 658,045 entries (filtered from 657,902 total entries)

Previous 1 2 3 4 5 … 65805 Next

And query them online





Still room for optimization

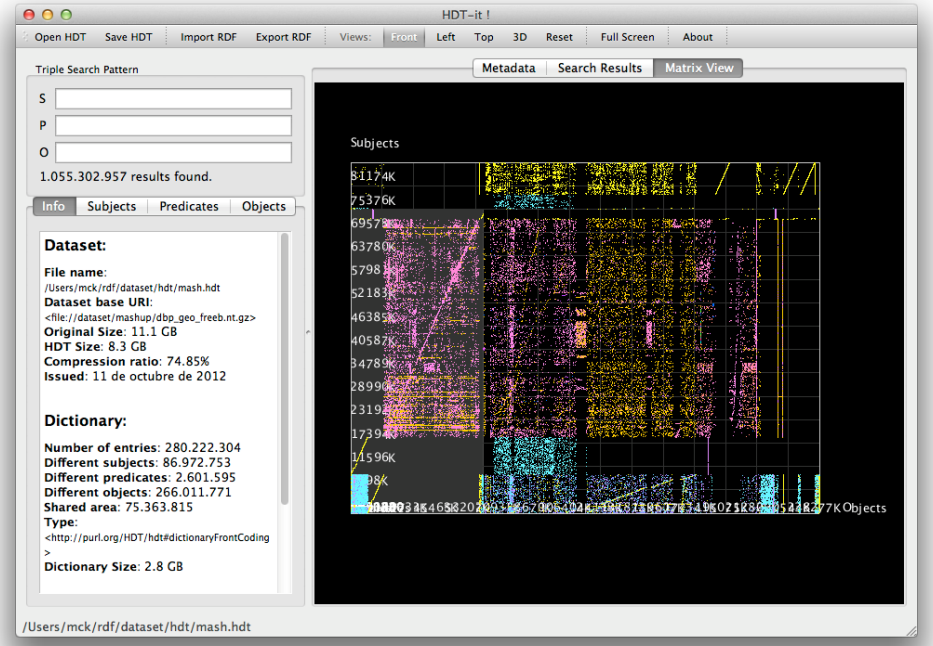
- *Dynamicity:*
  - **Most compact data structures are "static", but data may evolve**
  - **Tradeoff between compression and fast generation**
- *Advanced capabilities:*
  - **Reasoning (entailment)**
- *Query optimization:*
  - **Leverage precomputed statistics (stored in the Header)**
  - **Leverage on-demand statistics (Triple Pattern cardinality)**

How can I consume HDT

- GUI Tool
- C++/Java Libraries
  - Basic command line tools
  - Use them in your projects
- Set up a SPARQL Endpoint with HDT and Fuseki
- Set up a Linked Data Fragments Endpoint with HDT

Consuming HDT

- 1) Desktop tool HDT-it!
  - Thanks to Mario Arias



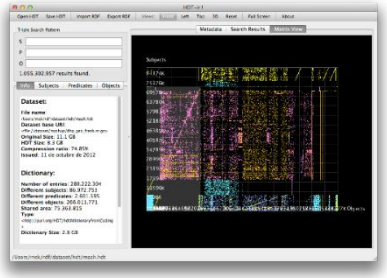
Consuming HDT

Try it yourself!

- 1) Desktop tool HDT-it!
  - Download the tool for your OS:
    - http://www.rdfhdt.org/downloads/
  - Get an HDT dataset from the web
    - http://www.rdfhdt.org/datasets/

    OR
    - http://lodlaundromat.org/wardrobe/

    OR convert your RDF dataset with the tool.



  - As a suggestion of small datasets:
    - SWDF (242K triples) or the bigger DBLP (55M triples)

Consuming HDT

- 2) C++/Java Libraries: https://github.com/rdfhdt
  - Command line Tools

	HDT-C++	HDT-Java
Basic command line tools	X	X
TP search	X	X
Full SPARQL	-	with Jena
Parametrizable Compression	X	-
Full text support	X	-
Practical Uses	LDF	Jena, Fuseki

Consuming HDT

Try it yourself!

- 2) C++/Java Libraries: https://github.com/rdfhdt
  - Command line Tools
    - For simplicity, we will use **Java** here
    - Download hdt-java library from https://github.com/rdfhdt/hdt-java/
      - git clone https://github.com/rdfhdt/hdt-java.git
    - Install the library with maven:
      - mvn install
    - Query an HDT file:
      - Go to HDT-cli and execute:
        - ./bin/hdhSearch.sh /path/to/your/hdt
        - This will open a simple console where you can query triple patterns
    - Export/Import
      - $> rdf2hdt file.nt output.hdt
      - $> hdt2rdf file.hdt output.nt

Consuming HDT

Try it yourself!

- 2) C++/Java Libraries: https://github.com/rdfhdt
  - Use in your projects
    - E.g. in java, include dependencies in pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.
  <modelVersion>4.0.0</modelVersion>
  <groupId>example.org</groupId>
  <artifactId>testHDT</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <dependencies>
    <dependency>
        <groupId>org.rdfhdt</groupId>
        <artifactId>hdt-java-core</artifactId>
        <version>2.0-SNAPSHOT</version> <!-- 1.1 is the published one, 2.0-SNAPSHOT is the one we have installed locally -->
    </dependency>
    <dependency>
        <groupId>org.rdfhdt</groupId>
        <artifactId>hdt-jena</artifactId>
        <version>2.0-SNAPSHOT</version> <!-- 1.1 is the published one, 2.0-SNAPSHOT is the one we have installed locally -->
    </dependency>
  </dependencies>
</project>
```

Consuming HDT

Try it yourself!

- 2) C++/Java Libraries: https://github.com/rdfhdt
  - Use in your projects
    - E.g. in java, include dependencies in pom.xml
      - Example 1: create a new Class and query our HDT

```java
package testHDT;

import java.io.IOException;

import org.rdfhdt.hdt.exceptions.NotFoundException;
import org.rdfhdt.hdt.hdt.HDT;
import org.rdfhdt.hdt.hdt.HDTManager;
import org.rdfhdt.hdt.triples.IteratorTripleString;
import org.rdfhdt.hdt.triples.TripleString;

public class FirstTest {

    public static void main(String[] args) throws NotFoundException,
    IOException {

        HDT myhdt = HDTManager.mapIndexedHDT("/path/to/your.hdt", null);
                            // use loadIndexedHDT if you have enough RAM

        IteratorTripleString it = myhdt.search("", "", "");
                            // any S,P,O. Use "" for a variable.

        while (it.hasNext()){
            TripleString triple = it.next();
            System.out.println(triple);
        }
    }

}
```

Consuming HDT

Try it yourself!

- 2) C++/Java Libraries: https://github.com/rdfhdt
  - Use in your projects
    - E.g. in java, include dependencies in pom.xml
      - Example 2: Load the model in Jena and query full SPARQL

```
18
19 public class JenaTest {
20
21     public static void main(String[] args) throws NotFoundException, IOException {
22
23         HDT myhdt = HDTManager.mapIndexedHDT("/home/javi/LearningWeek/data/swdf-2012-11-28.hdt", null);
24         // use loadIndexedHDT if you have enough RAM
25
26         // Create Jena Model on top of HDT.
27         HDTGraph graph = new HDTGraph(myhdt);
28         Model model = ModelFactory.createModelForGraph(graph);
29
30         String queryString = "SELECT ?s ?p ?o WHERE {?s <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> . "
31                 + "?s ?p ?o }";
32         Query query = QueryFactory.create(queryString);
33         QueryExecution qexec = QueryExecutionFactory.create(query, model);
34         ResultSet results = qexec.execSelect();
35         for (; results.hasNext();) {
36             QuerySolution soln = results.nextSolution();
37             RDFNode s = soln.get("s"); // Get a result variable by name.
38             RDFNode p = soln.get("p"); // Get a result variable by name.
39             RDFNode o = soln.get("o"); // Get a result variable by name.
40
41             System.out.println("Solution " + s + " " + p + " " + o);
42         }
43     }
44 }
```

Consuming HDT

Try it yourself!

- 3) Set up a SPARQL Endpoint with HDT and Fuseki
  - Go to hdt-fuseki and compile adding the dependencies:
    - mvn package dependency:copy-dependencies
  - Run fuseki
    - ./bin/hdtEndpoint.sh --hdt path/to/dataset.hdt /mydataset
  - Open your Web Browser and go to: http://localhost:3030
    - Select Control Panel / Dataset / myDataset and click Select
    - Type your SPARQL Query and see the results.
  - Be careful with the number of results, here there is no limitation in the number of results such as in e.g. virtuoso:
    - select * WHERE{ ?s ?p ?o} LIMIT 400

Consuming HDT

- 4) Set up a Linked Data Fragments Endpoint with HDT
  - Download LDF Server (Node.js is the best one but we will use java for simplicity in the installation).
    - git clone https://github.com/LinkedDataFragments/Server.Java.git

  or download https://github.com/LinkedDataFragments/Server.Java/archive/master.zip
  - Install the server, avoid the test (it fails :)
    - mvn install -Dmaven.test.skip=true
  - Open the file config-example.json and modify the settings to point to your hdt, e.g.
    - "settings": { "file": "/home/user/myfile.hdt" }
  - Run the server
    - java -jar target/ldf-server.jar
  - Access http://localhost:8080

ACKs


DataWebResearch
http://dataweb.infor.uva.es


Universidad de Valladolid


Insight


UNIVERSIDAD DE CHILE


#LD
Linked Data Fragments


GHENT
UNIVERSITY


imec


HPI
Hasso
Plattner
Institut
IT Systems Engineering | Universität Potsdam


VU
UNIVERSITY
AMSTERDAM


Knowledge Representation
and Reasoning