Lanedo

Tracker

A crash course

About us

- Aleksander Morgado

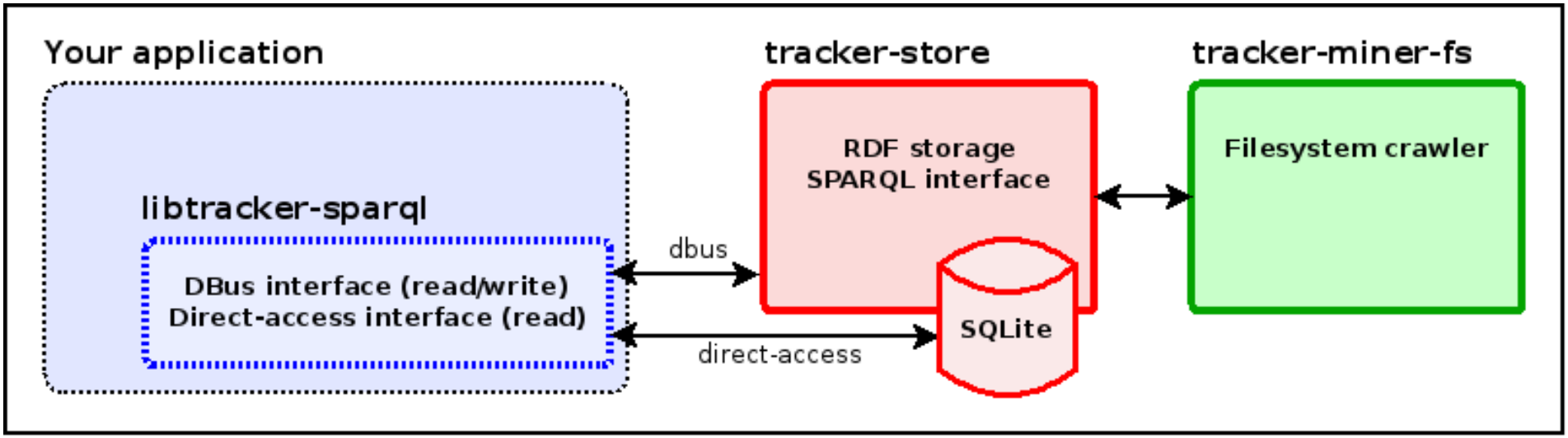
  aleksander@lanedo.com

- Carlos Garnacho

  carlos@lanedo.com

Tracker, in pieces

- Structured storage
- Data miners



Structured storage

- Information is a graph [1]
  - Vertices are objects
  - Edges establish relations
  - RDF for the ones familiar with it

[1] A labeled directed multi-graph for the picky

Structured storage (II)

- Insertions
  - Fundamentally a set of triples of data: subject / predicate / object
    - Subject = object → Vertices
    - Predicate → Edge

Example 1.

```
INSERT {
  <fooBar> a              nie:InformationElement .
  <fooBar> dc:creator "The king of the world" .
  <fooBar> dc:date    "2013-07-24T09:45:09Z" .
  <fooBar> nie:plainTextContent "Lorem ipsum..."
}
```

Example 2.

```
INSERT {
  <fooBar> a                nie:InformationElement ;
           dc:creator “The king of the world” ;
           dc:date    “2013-07-24T09:45:09Z” ;
           nie:plainTextContent “Lorem ipsum...”
}
```

Structured storage (III)

- Queries (`SELECT … WHERE { … }`)
  - The Select clause let you define the returned data
  - The Where clause let you define a minimal graph,
    - all returned items satisfy it
  - Named variables (starting with '?')
    - Act as placeholders on the Select clause
    - Act as the '*' wildcard on the Where clause

Example 3.

```
SELECT ?creator WHERE {
  <fooBar> dc:creator ?creator .
}
```

Example 4.

```
SELECT ?predicate ?object WHERE {
  <fooBar> ?predicate ?object .
}
```

Example 5.

```
SELECT ?subject ?object WHERE {
  ?subject dc:creator ?object .
}
```

Structured storage (IV)

- Query filters
  - `SELECT … WHERE { … }` as shown only matches graphs, doesn't evaluate content
  - Filters let you apply arbitrary restrictions on the content
    - Comparisons
    - Substring matching
    - ...

Example 6.

```
SELECT ?subject ?object WHERE {
  ?subject dc:creator ?object .
  FILTER (fn:starts-with (?object, 'The king')) .
}
```

Example 7.

```
SELECT ?subject WHERE {
  ?subject fts:match 'ipsu*' .
}
```

Structure of data (O#*%logy)

- A predefined, comprehensive set of schemas
  - Separated by domains of data, visible as prefixes in object definitions
- Defined on the same terms than data itself
- Both objects and relations between them are subject to inheritance and more specific definitions

Example 8.

```
SELECT
  ?shortname ?prefix
WHERE {
  ?prefix a                  tracker:Namespace ;
          tracker:prefix ?shortname .
}
```

Schemas

- xsd: Basic data types
- rdf/rdfs: Resources and properties
- dc: Common set of superproperties
- nie: Topmost classes
- nao: Tags, ratings and other annotations
- nco: Contacts
- nfo: Files and local resources
- nmo: Messaging
- ncal / scal: Calendars and events
- nmm: Multimedia objects
- mfo: Feeds
- mtp: Media transfer
- tracker: Tracker additions
- slo: geolocation

Example 9.

```
SELECT
   ?predicate ?object
WHERE {
   rdfs:Resource ?predicate ?object .
}
```

Example 10.

```
SELECT
  ?resource
WHERE {
  ?resource rdfs:subClassOf rdfs:Resource .
}
```

Example 11.

```
SELECT
  ?resource ?subclass
WHERE {
  ?resource rdfs:subClassOf ?subclass .
}
```

Example 12.

```
SELECT
  ?property
WHERE {
  ?property rdfs:domain rdfs:Resource .
}
```

Example 13.

```
SELECT
   ?property ?resource
WHERE {
   ?property rdfs:domain ?resource .
}
```

Example 14.

```
SELECT
   ?property ?parent
WHERE {
   ?property rdfs:subPropertyOf ?parent .
}
```

Thanks!

- Website:

  `http://projects.gnome.org/tracker`

- Wiki:

  `https://wiki.gnome.org/Tracker`

- Mailing list:

  `tracker-list (at) gnome.org`