RDF úložiště a SPARQL

Reprezentace a zpracování znalostí na WWW
29. 10. 2012

Jindřich Mynarz

RDF úložiště

- Triple store: trojice
  - subjekt predikát objekt
- Quad store: čtveřice
  - graf subjekt predikát objekt
- Back-end
  - soubor, v paměti, relační databáze...
- Indexování
  - Permutace trojic/čtveřic
  - Dodatečné plnotextové indexování
- Inferencování
  - Materializace inferencí

RDF úložiště

- Optimalizace
  - Volba plánu pro vykonání dotazu
  - Přepisování SPARQL dotazů
- Škálování
  - Podpora ukládání miliard trojic
  - Komprese
  - Clustery

Implementace

- Virtuoso (OpenLink)
- 4store (Garlik)
- Stardog (Clark & Parsia)
- Jena TDB (Apache Software Foundation)
- Sesame (Aduna)
- BigOWLIM (Ontotext)
- AllegroGraph (Franz Inc.)
- Oracle 11g (Oracle)
- Meronymy SPARQL Database Server (Meronymy)

Více...

SPARQL

- Rodina standardů pro práci s RDF
  - SPARQL Query: dotazování
  - SPARQL Update: modifikace
  - SPARQL Protocol: komunikační protokol
  - SPARQL Query Results XML Format: serializace výsledků SPARQL dotazu

Anatomie SPARQL dotazu

- Komentář
  - `# komentář`
- Prolog
  - Deklarace **prefixů** použitých jmenných prostorů
  - `PREFIX prefix: <http://jmenny.prostor/>`
- Typ dotazu
  - `SELECT`: vrací navázané proměnné
  - `CONSTRUCT`: vrací RDF graf
  - `ASK`: vrací, zdali pro dotaz existuje řešení
  - `DESCRIBE`: vrací popis daného zdroje

Anatomie SPARQL dotazu

- Modifikátor řešení
  - `DISTINCT`: pouze unikátní řešení
  - `REDUCED`: duplicitní řešení mohou být odstraněna
- Řešení
  - Navázané proměnné (`?promenna`, `*`)
  - Grafový vzor
- Určení dotazovaných dat
  - `FROM`: přidává trojice z pojmenovaného grafu do výchozího grafu
  - `FROM NAMED`: přidává trojice z pojmenovaného grafu, jehož URI musí být v grafovém vzoru explicitně určeno pomocí `GRAPH`

Anatomie SPARQL dotazu

- Podmínka
  - `WHERE { ` grafový vzor ` }`
  - Grafový vzor (subgraf) je **příklad** hledaných dat
  - Syntaxe vzoru
    - Turtle
    - Proměnné: `?promenna`
    - `GRAPH`: určení pojmenovaného grafu
    - `UNION`: sjednocení alternativních vzorů
    - `OPTIONAL`: nepovinný vzor
    - Poddotazy: použití řešení vnořených dotazů ve vzoru

Anatomie SPARQL dotazu

- Podmínka: filtr
  - `FILTER`
  - Logický výraz, který odfiltruje řešení, pro něž nabývá hodnoty `false`.
  - 3-hodnotová logika (`true`, `false`, `error`)
  - Funkce
    - vracející logickou hodnotu, např. `isURI()`, `BOUND()`, `REGEX()`

Anatomie SPARQL dotazu

- Modifikátory dotazu
  - `LIMIT počet`: omezení počtu řešení
  - `OFFSET pořadí`: stanovení pořadí prvního vybraného řešení ve všech řešeních
  - `ORDER BY`: seřazení řešení
    - Vzestupně: `ORDER BY ASC(?proměnná)`
    - Sestupně: `ORDER BY DESC(?proměnná)`

Anatomie SPARQL dotazu

- SPARQL 1.1 Query
  - Negace: `NOT EXISTS`, `MINUS`
  - Agregace: `GROUP BY`, `HAVING`, `SUM`, `COUNT`, `AVG`
  - Federace: `SERVICE`
  - Vnořené dotazy
  - Přiřazování proměnných: `BIND`
  - `...`
- SPARQL 1.1 Update
  - `INSERT, DELETE, LOAD, CLEAR...`

Příklady

```
# 10 libovolných trojic

SELECT * WHERE {
   ?s ?p ?o .
}
LIMIT 10
```

Příklady

```
# Všechny unikátní typy zdrojů

SELECT DISTINCT ?class WHERE {
  [] a ?class .
}
```

Příklady

```
# Všechny unikátní predikáty zdrojů

SELECT DISTINCT ?property WHERE {
  [] ?property [] .
}
```

Příklady

```
# Celkový počet trojic

SELECT (COUNT(*) AS ?
numberOfTriples)
WHERE {
   ?s ?p ?o .
}
```

Příklady

```
# 10 trojic z daného grafu

SELECT * FROM NAMED <http://ld.opendata.
cz/resource/dataset/czso.cz/demography>
WHERE {
  GRAPH <http://ld.opendata.
cz/resource/dataset/czso.cz/demography> {
    ?s ?p ?o .
  }
} LIMIT 10
```

Příklady

```
# dc:title a zároveň dcterms:title

PREFIX dc: <http://purl.org/dc/elements/1.
1/>
PREFIX dcterms: <http://purl.org/dc/terms/>

SELECT ?title WHERE {
  { ?s dc:title ?title . }
  UNION
  { ?s dcterms:title ?title . }
}
```

Příklady

```
# Negation as failure (SPARQL 1.0)

PREFIX skos: <http://www.w3.
org/2004/02/skos/core#>
SELECT ?concept WHERE {
  ?concept a skos:Concept .
  OPTIONAL {
    ?concept skos:broader ?broaderConcept .
  }
  FILTER(!BOUND(?broaderConcept))
}
```

Příklady

```
# Negace (SPARQL 1.1)

PREFIX skos: <http://www.w3.
org/2004/02/skos/core#>

SELECT ?concept WHERE {
  ?concept a skos:Concept .
  FILTER NOT EXISTS {
    ?concept skos:broader ?broaderConcept .
  }
}
```

Nástroje

- Prefix.cc: vyhledávání jmenných prostorů pro populární prefixy
- SPARQL Query Validator
- SPARQLer: generický SPARQL procesor
- SPARQL2NL: překlad SPARQL dotazů do přirozeného jazyka
- SPARQLed: interaktivní editor SPARQL dotazů
- SPARQLbin: sdílení SPARQL dotazů
- SPARQL Endpoint Status: monitorování SPARQL endpointů

Příklady SPARQL endpointů

- DBpedia
- Veřejné zakázky
- British National Bibliography
- DataGovIE - Irish Government Data
- WordNet (RKBExplorer)
- Katalog dat Datahub.io

Zdroje

- Bob DuCharme. Learning SPARQL.
- Lee Feigenbaum; Eric Prud'hommeaux. SPARQL by example.
- Eric Prud'hommeaux; Andy Seaborne. SPARQL Query Language for RDF.