Triple Stores

What is a triple store?

- A database for RDF triples

- Can ingest RDF in a variety of formats

- Supports a query language
  - SPARQL is the W3C recommendation
  - Other RDF query languages exist (e.g., RDQL)
  - Might or might not do inferencing

- Triples stored in memory or in a persistent backend

- Persistence provided by a relational DBMS (e.g., mySQL) or a custom DB for efficiency

Architectures

Can be divided into several categories: *In-memory, Native store, Non-native store*

- In memory: RDF Graph is stored as triples in main memory

- Native store: Persistent storage systems with custom DBs, e.g.: JENA TDB2, Virtuoso, AllegroGraph, Oracle 11g, Amazon Neptune

- Non-Native store: Persistent storage systems set-up to run on third party DBs, e.g., Jena SDB using mysql or postgres

Architecture trade-offs

- In memory is fastest, obviously, but load time has to be factored in

- Native stores are fast, scalable, and popular now

- Non-native stores may be better if you have a lot of updates and/or need good concurrency control

- See the W3C page on large triple stores for some data on scaling for many stores

Large triple stores in 2022

Ordered by reported counts of triples in descending order

Quads, Quints and Named Graphs

- Many triple stores support quads for named graphs

- A named graph is just an RDF with a URI name often called the *context*

- Such a triple store divides its data a default graph and zero or more additional named graphs

- SPARQL has support for named graphs

- De facto standards exist for representing quad data, e.g., n-quads and TriG (a turtle/N3 variant)

- AllegroGraph stores quints (S,P,O,C,ID), the ID can be used to attach metadata to a triple
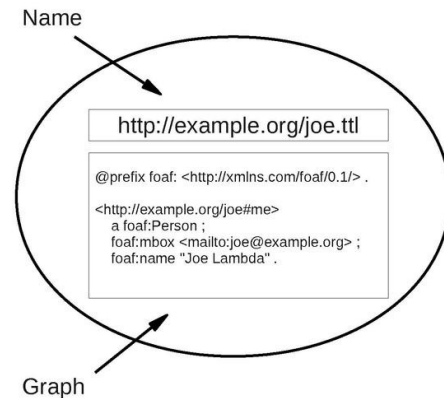
Support for Reasoning

- Many triple stores don't do much (or any) reasoning and use a simple model:
  - You do the reasoning to materialize the triples you want, which you then load into the store
  - Triple store provides query and update APIs, access control, SPARQL interface, efficient indexing, etc.

- Some do support reasoning, e.g.,
  - Jena has a native rules engine and an API for external reasoners (e.g., Pellet, Fact++)
  - Stardog supports OWL DL reasoning via query expansion and other efficient techniques
  - RDFox support Datalog-like rules and truth maintenance

Example: Jena

- An open-source Java system originally developed by HP (2002-2009)
    - Moved to Apache when HP Labs discontinued its Semantic Web research program
    - https://jena.apache.org/
- Using TDB2 native store, handles ~17B triples
- Good tutorials and documentation
- Has internal reasoners and can work standards compliant reasoners such as Pellet
- Supports a Native API and SPARQL via Fuseki

Example: rdf4j

- [rdf4j](#) (nee Sesame): open-source RDF framework supporting RDFS inference & querying

- https://rdf4j.org/

- Implemented in Java

- Query languages: SPARQL and others

- Triples: stored in memory, on disk, or in RDBMS

- Has a native RDFS reasoner

- Easy to setup & use, but tops out at ~70M triples

Example: Stardog

- http://stardog.com/ by Clark and Parsia
- Pure Java RDF database ("quad store")
- Lightweight and very fast for in-memory use
- Reasoning support via Pellet for OWL DL and query rewriting for OWL 2 QL, EL & RL
- Command line interface and JAVA API
- Powerful Web browser interface for database management and queying
- Commercial, but has good free systems

Summary

- A triple store is an essential component of any system using RDF

- There are a number of good ones available, both open sourced and commercial

- Developing triple stores for large-scale parallel systems is still a research topic