Vorprojekt

Verantwortlicher: Daniel Alexander

Daniel Alexander, Jakob Kusnick, Richard Khulusi, Christopher Schott, Tobias Turke, Nikolas Oberhäuser, Hans Dieter Pogrzeba, Felix Conrads

1. Vorprojekt - Aufgabe

Als Vorprojekt soll ein umfangreicher Benchmark unterschiedlicher Triplestores durchgeführt werden. Dabei sollen typische Queries abgefragt werden, wie sie vom späteren Nutzer zu erwarten sind.

Für jeden der Triplestores wird dabei eine Testinstallation durchgeführt und ein Testdatenbestand von über einer Million Partien geladen. Anschließend werden die Queries auf 10%, 50% und 100% des Datenbestands abgefragt, wobei einzelne Parameter fortlaufend verändert werden, um ein Abrufen der Ergebnisse aus dem Cache zu vermeiden.

Als Ergebnis sollen die Leistungen der Triplestores in einer Übersicht grafisch aufbereitet und für das Projekt analysiert werden. Abschließend wird auf Basis der Analyse ein Triplestore für das Projekt ausgewählt.

2. Begründung

Die verschiedenen Triplestores legen unterschiedliche Schwerpunkte in Bezug auf Effizienz und Geschwindigkeit. Damit die späteren Anfragen möglichst performant bearbeitet werden können, soll geklärt werden, welcher Triplestore sich für den konkreten Anwendungsfall am besten eignet.

Das Hauptkriterium ist hierbei der schnelle Lesezugriff, um die Nutzer nicht unnötig warten zu lassen. Weniger wichtig ist die Geschwindigkeit beim Schreiben, da nur einmalig beim ersten Crawler-Suchlauf eine große Menge Daten in den Triplestore geschrieben wird und danach nur noch vereinzelt Daten hinzugefügt werden. Weiterhin ist es nötig, dass der Triplestore auch bei der Verwaltung großer Datenmengen effizient bleibt.

Durch das Vorprojekt findet bereits eine Einarbeitung in den Umgang mit Triplestores sowie die vorhandenen Schnittstellen der konkreten Triplestores statt. Um den Benchmark automatisiert durchführen zu können, wird jeweils eine Anbindung zur Dateneingabe geschrieben oder ggf. ein durch den Triplestore bereitgestelltes Framework genutzt. Somit wird die Art der Dateneingabe und -ausgabe der letztendlich verwendeten Datenbank bereits vor Beginn des Hauptprojekts geklärt.

3. Auswahl der zu testenden Triplestores

Die folgende Liste wurde auf Basis einer Recherche ermittelt, wobei dabei im Vordergrund stand, Triplestores zu finden, welche kostenfrei zur Verfügung stehen und die geforderten Datenmengen handhaben können. Weiterhin wurde darauf Wert gelegt, dass Schnittstellen über Frameworks oder eigene APIs zur Verfügung stehen. Zusätzlich sollten die Triplestores aus leistungstechnischen Gründen auf einem Linuxserver installierbar sein, da die Performance etwa durch virtuelle Maschinen enorm leidet.

Name	**Schnittstellen / Frameworks / Formate**	**Mögl. Größe (Triple)**
BigData	SPARQL	12,7B
AllegroGraph	Jena	1T
Virtuoso	SQL client Librarys, SPARQL (JDBC Access), Turtle und RDF/XML	15,4B
4Store	SPARQL Endpoint mit SPARQL Update, eigene API (nur über Maven nutzbar)	15B
OWLIM	Jena, Sesame, SPARQL	12B

Zu testende Triplestores

4. Queries und Nutzerszenarien

Es werden ca. 50 Queries, welche auf sinnvollen Nutzerszenarien basieren, ausgewählt und nach statistischer Analyse auf Relevanz gewichtet.
In der DBpedia existieren Statistiken, wie viele Parameter je Anfrage vorkommen. Die Auswertung der Statistik nach Anzahl der Parameter ermöglicht eine Gewichtung der Queries.
Die Queries stehen auf http://pcai042.informatik.uni-leipzig.de/~swp14-cci/