

Performanzsteigerung datenbankgestützter RDF-Triple-Stores

Michael Martin

`martin@informatik.uni-leipzig.de`

Abstract: Die Benutzung von RDF-Triple-Stores im Semantic Web ist bei der Verwendung großer Wissensbasen fast unumgänglich. Hierzu hat sich unter anderem der Einsatz von Datenbankmanagementsystemen für grundlegende Speichermechanismen etabliert. Allerdings sind die Performanz- und Skalierbarkeitseigenschaften von RDF-Triple-Stores im Gegensatz zu herkömmlichen relationalen Datenbanken noch nicht optimal. Dies erschwert die Verbreitung von Technologien des Semantic Web sowie eine damit verbundene Entwicklung semantischer Web-Applikationen, bei denen der Fokus auf schnelle Datenverarbeitung gelegt wird. In dieser Arbeit wird das Forschungsvorhaben zur Behebung dieses Problems vorgestellt.

1 Einleitung

Die Entwicklung von im *World Wide Web* (WWW) eingesetzten Technologien ist längst nicht abgeschlossen. Gerade die Erweiterung des WWW um eine semantische Schicht (*Semantic Web*) ist einer der aktuellen Forschungsschwerpunkte. Im Semantic Web werden unter anderem Wissensbasen auf Grundlage des *Resource Description Framework / Schema* (RDF/S) [Bec04, BG04] und der *Ontology Web Language* (OWL) [PSHH04] eingesetzt. Mittels dieser eigens für das Semantic Web entwickelten Technologien ist es möglich, komplexe Informationsstrukturen und dazugehörige Interpretationsvorschriften zu modellieren und maschineninterpretierbar zu speichern. Zur Serialisierung und Deserialisierung derart gespeicherter Informationen können verschiedene semantische Technologien eingesetzt werden. Einerseits ist es möglich diese Modelle mit verschiedenen Notationsformen wie RDF/XML, N3 [BLC08] oder Turtle [BBL08] zu kodieren und zu speichern. Andererseits können datenbankgestützte *RDF-Triple-Stores* verwendet werden, die aufgrund des Einsatzes von Datenbanken als Basis-Speicherverfahren den Umgang mit großen Datenmengen beherrschen. Die Auswahl eines geeigneten RDF-Triple-Stores beziehungsweise eines dafür benötigten Datenbankschemas kann auf verschiedenen Aspekten basieren [Gro03]. Werden beispielsweise häufig SPARQL-Anfragen [CFT08] mit vorhandenem Prädikat an einen RDF-Triple-Store gestellt, ist es vorteilhaft ein auf eigenschaftsorientierten Konzepten basierendes Datenbankschema zu verwenden. Besteht die Möglichkeit objektrelationale Datenbanken (ORDBMS) zum Einsatz zu bringen, können auch objektrelationale Konzepte zur Speicherung verwendet werden, um beispielsweise Konzepthierarchien auf Tabellenhierarchien abzubilden. Beide Ansätze dienen unter anderem der Performanzsteigerung bei Anfrage- und Manipulations-Prozessen. Designent-

scheidungen unter diesem Kriterium sind notwendig, da mangelnde Performanz und Skalierbarkeit von Speicherungsoperationen und Anfragen an die im RDF-Datenmodell repräsentierten Wissensbasen entscheidende Hindernisse zur breiten Anwendung von semantischen Technologien darstellen. Gerade bei der Entwicklung semantischer Web-Applikationen kann auf die Verwendung performanterer RDF-Triple-Stores nur selten verzichtet werden. Hierzu wurden bisher enorme Anstrengungen unternommen, um effizientere Persistenz-, Indizierungs- und Query-Optimizing-Strategien zu implementieren (z.B. OpenLink's Virtuoso, Oracle 11G, OWLIM/Sesame), die allerdings für hochgradig skalierbare oder massiv parallel anfragbare Wissensbasen (insbesondere im Vergleich mit relationalen Datenbanken) noch unzureichend sind. Eine aus dem Datenbankbereich bekannte Strategie zur weiteren massiven Performanzsteigerung sind *View-Maintenance*-Algorithmen [ZGMHW95], bei denen die Ergebnisse häufig auftretender Anfragen zwischengespeichert (*View-Caching*) und wiederverwendet werden. Durch die fundamentalen Unterschiede zwischen dem relationalen Datenmodell (Tabellen) und dem RDF-Datenmodell (Triple) lassen sich existierende View-Maintenance-Algorithmen aus dem Datenbankbereich nicht einfach übertragen. Weiterhin existiert derzeit noch kein formales Modell, mit Hilfe dessen die Verwendung von RDF-Triple-Stores unter Benutzung derartiger performanzsteigernder Methoden verbessert werden kann. An dieser Stelle wird mit dem Forschungsvorhaben angesetzt und im folgenden der Stand der Technik in Kapitel 2 und das avisierte Ziel in Kapitel 3 vorgestellt. Weiterhin wird der aktuelle Stand der Arbeit in Kapitel 4 skizziert und eine Zusammenfassung in Kapitel 5 gegeben.

2 Stand der Technik und verwandte Arbeiten

Wie in der Einleitung erwähnt, existieren Implementierungen in Form von Bibliotheken und Frameworks für RDF-Triple-Stores, mittels derer Anfrage- und Manipulationsoperationen von anderen (Web-)Applikations-Komponenten abstrahiert werden können. Um die zu erstellenden performanzsteigernden Methoden zum Einsatz zu bringen, muss beispielsweise an den in Tabelle 1 aufgeführten Implementierungen angesetzt werden. Derartige Implementierungen werden beispielsweise im OntoWiki [ADR06], einem auf Wiki-Ansätzen basierende und unter anderem als Ontologie-Editor konstruierte semantische Web-Applikation, eingesetzt. OntoWiki würde von der Integration performanzsteigernder Methoden profitieren und könnte zudem auch zur Erruierung der Funktionstüchtigkeit der zu implementierenden Methoden eingesetzt werden. Weitere Web-Applikationen, die für Analyse, Entwurf und Test von Funktionalitäten benutzt werden können, sind DBpedia [ABL⁺07], in der aus Wikipedia extrahierte und in RDF transformierte Daten über ein Web-Interface beziehungsweise über einen SPARQL-Endpunkt abgerufen werden können, und das niederländische Tourismusportal vakantieland.nl [Mar07]. Verwandte Arbeiten zur Optimierung von RDF-Triple-Stores unter den Kriterien der Performanz und Skalierbarkeit sind [HDS05], [MTCP04], [HD05] und [VOS02]. Die darin enthaltenen Beschreibungen und ermittelten Erkenntnisse bieten eine gute Basis zur Erstellung des in Kapitel 3 avisierten Ergebnisses.

Name	Beschreibung
<i>ARC</i>	leichtgewichtiger in PHP implementierter SPARQL-Server, der direkt auf einer MYSQL-Datenbank arbeitet
<i>Jena</i>	in Java implementierte(s) Framework/Bibliothek zum Erzeugen semantischer Web-Applikationen, enthält Umgebung für RDF/S, OWL, SPARQL, GRDDL und eine regelbasierte <i>inference engine</i> . Ist auch benutzbar als RDF-Datenbank über den Joseki Layer
<i>OpenLink Virtuoso</i>	in C++ implementierter Web-Applikations-Server und ORDBMS, der SQL, XML und RDF-Management unterstützt, sowie Support für SPARQL, ODBC, GRDDL, JDBC, ADO.NET, WebDav etc. leistet.
<i>RDF-API for PHP (RAP)</i>	In PHP implementierte Klassenbibliothek zur Manipulation von RDF-Modellen und Parsen/Serialisieren von RDF
<i>Redland librdf</i>	In C implementiertes Framework mit Sammlung von Bibliotheken für RDF-Unterstützung. Unterstützt auch Turtle, N3, Atom, RSS und enthält eine SPARQL- und GRDDL-Implementierung
<i>Sesame</i>	In Java und Python implementiertes Framework mit RDF-Datenbank und Bibliothek zur RDF-Unterstützung
<i>YARS</i>	In Java implementierter performanter RDF-Store, in dem N3 zur Speicherung und für Abfragen (N3QL) genutzt wird.

Tabelle 1: Implementierungen für RDF-Triple-Stores (<http://esw.w3.org/topic/SemanticWebTools>)

3 Avisiertes Ergebnis der Arbeit

Einen Ansatz zur Lösung des in Kapitel 1 dargestellten Problems ist Gegenstand dieser Forschungsarbeit. Mittels eines zu erstellenden formalen Modells werden Vorgehensweisen verankert, die durch verschiedene View-Maintenance-Algorithmen realisiert werden. Hierbei werden einerseits bestehende Algorithmen erruiert und insofern möglich auf die Spezifika von RDF-Triple-Stores angepasst, und andererseits, falls eine Übertragung nicht möglich ist, eigens Algorithmen geschaffen. Das somit auszuarbeitende abstrakte View-Maintenance-Modell wird unter anderem die Elemente *Tripelpersistenz*, *Anfragen*, *Latenz* sowie *Ausführungs-* und *View-Materialisierungsgeschwindigkeit* beschreiben. Weiterhin werden Heuristiken für das View/Cache-Management sowie regelbasierte Mechanismen zur Query-Subsumption und Cache-Object-Invalidation erarbeitet. Das geschilderte Problem erfährt eine besondere Erschwernis durch die in RDF oder höheren auf RDF basierenden Wissensrepräsentationsformalismen (Taxonomien/RDFS oder Ontologien/OWL) kodierten impliziten Informationen. Diese kodierbaren Zusammenhänge, die nur durch Inferenz (Reasoning) aufgelöst/ermittelt werden können, müssen von dem zu definierenden formalen Modell und den zu entwickelnden Algorithmen berücksichtigt werden. Somit ergibt sich die folgende Methodologie zur Abarbeitung des beschriebenen Vorhabens.

1. **Analyse und Entwurf:** Entwicklung von Heuristiken für das View/Cache-Management

2. **Analyse und Entwurf:** Erarbeitung regelbasierte Mechanismen für Query-Subsumption und Cache-Object-Invalidation unter Berücksichtigung der Besonderheiten von RDF (Kodierung von implizit vorhandenem Wissen)
3. **Entwurf und Design:** Erarbeitung des abstrakten View-Maintenance-Modells für RDF-Triple-Stores
4. **Design und Implementierung:** Implementierung einer leichtgewichtigen Abstraktionsschicht (Proxy) über beliebigen SPARQL-Endpunkten
5. **Implementierung und Test:** Evaluation und Validierung mit bestehenden RDF-Triple-Stores (vgl. Kapitel 2) unter Verwendung spezieller Benchmark-Spezifikationen (wie beispielsweise [BS08])

4 Aktueller Stand

Auf Basis bisheriger Entwicklungen sowie der Verwendung semantischer Web-Applikationen, wie beispielsweise dem OntoWiki-Projekt und vakantieland.nl, wurde das in dieser Ausarbeitung beschriebene Problem identifiziert und der Bedarf an im Kapitel 3 beschriebenen Entwicklungen ermittelt. Allerdings befindet sich das Forschungsvorhaben noch in der Planungs-/Konkretisierungsphase, in welcher unter anderem eine nähere Analyse des beschriebenen Lösungsansatz (vgl. 3) durchgeführt wird.

5 Zusammenfassung und Ausblick

In dieser Arbeit wurde das Problem mangelnder Performanz und schlechten Skalierbareigenschaften von RDF-Triple-Stores durchgeführt und ein Lösungsansatz sowie die Strategie zur Behebung dieser Umstände gegeben. Hierzu wurde das Problem in Kapitel 1 und der Bedarf an Forschungsarbeit beschrieben. Weiterhin wurde der Stand der Technik in Kapitel 2 und eine damit verbundene Auswahl an bestehenden Implementierungen vorgestellt. Ein Überblick über das avisierte Ziel und den aktuellen Stand wurden in Kapitel 3 und 4 gegeben.

Nach Abschluss weiterer Analysen und darauf basierenden Designentscheidungen werden in anschließenden Ausarbeitungen konkrete Teile des zu entwickelnden formalen Modells sowie (diese Formalismen realisierende) Algorithmen der erwähnten Abstraktionsschicht vorgestellt. Die Funktionstüchtigkeit und die damit verbundenen Performanzgewinne werden unter anderem mittels der in Kapitel 2 und 3 skizzierten Implementierungen ermittelt.

Literatur

- [ABL⁺07] Sören Auer, Chris Bizer, Jens Lehmann, Georgi Kobilarov, Richard Cyganiak und Zachary Ives. DBpedia: A Nucleus for a Web of Open Data. In *Proc. of ISWC/ASWC2007*, Jgg. 4825 of *LNCS*, Seiten 715–728, Berlin, Heidelberg, November 2007. Springer Verlag.
- [ADR06] Sören Auer, Sebastian Dietzold und Thomas Riechert. OntoWiki - A Tool for Social, Semantic Collaboration. In *Proc. of 5th ISWC 2006*, Jgg. 4273 of *Lecture Notes in Computer Science*, Seiten 736–749. Springer, 2006.
- [BBL08] David Beckett und Tim Berners-Lee. Turtle - Terse RDF Triple Language. W3c team submission, W3C, January 2008.
- [Bec04] David Beckett. RDF/XML Syntax Specification (Revised). W3c recommendation, W3C, February 2004.
- [BG04] Dan Brickley und Ramanatgan V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3c recommendation, W3C, February 2004.
- [BLC08] Tim Berners-Lee und Dan Connolly. Notation3 (N3): A readable RDF syntax. W3c team submission, W3C, January 2008.
- [BS08] Chris Bizer und Andreas Schultz. Berlin SPARQL Benchmark (BSBM) Specification, July 2008. <http://www4.wiwi.fu-berlin.de/bizer/BerlinSPARQLBenchmark/spec/>.
- [CFT08] Kendall Grant Clark, Lee Feigenbaum und Elias Torres. SPARQL Protocol for RDF. W3c recommendation, W3C, January 2008.
- [Gro03] Jan Große. Speicherverfahren und Werkzeuge für RDF/S. XML Clearinghouse Report 6, Herausgeber: Prof. Dr. Ing. Robert Tolksdorf (Freie Universität Berlin) und Dr. Rainer Eckstein (Humboldt Universität Berlin), Dezember 2003.
- [HD05] Andreas Harth und Stefan Decker. Optimized Index Structures for Querying RDF from the Web. In *LA-WEB '05: Proceedings of the Third Latin American Web Congress*, Seite 71, Washington, DC, USA, 2005. IEEE Computer Society.
- [HDS05] Edward Hung, Yu Deng und V. S. Subrahmanian. RDF Aggregate Queries and Views. In *ICDE*, Seiten 717–728. IEEE Computer Society, 2005.
- [Mar07] Michael Martin. Exploring the Netherlands on a Semantic Path. In *CSSW*, Jgg. 113 of *LNI*, Seiten 179–. GI, 2007.
- [MTCPO4] Aimilia Magkanaraki, Val Tannen, Vassilis Christophides und Dimitris Plexousakis. Viewing the semantic web through RVL lenses. *J. Web Sem.*, 1(4):359–375, 2004.
- [PSHH04] Peter F. Patel-Schneider, Patrick Hayes und Ian Horrocks. OWL Web Ontology Language - Semantics and Abstract Syntax. W3c recommendation, W3C, 10 feb 2004.
- [VOS02] Raphael Volz, Daniel Oberle und Rudi Studer. On Views in the Semantic Web. In *2nd International Workshop on Databases, Documents and Information Fusion (DB-FUSION02)*, Karlsruhe, Germany, 07 2002.
- [ZGMHW95] Yue Zhuge, Héctor García-Molina, Joachim Hammer und Jennifer Widom. View maintenance in a warehousing environment. In *Proc. of 1995 ACM SIGMOD*, Seiten 316–327, New York, NY, USA, 1995. ACM.