Optimierung von Analytischen Abfragen über Statistical Linked Data mit MapReduce

Sébastien Jelsch[1], Benedikt Kämpgen[1] und Stefan Igel[2]

[1] FZI Forschungszentrum Informatik
sebastien.jelsch@fzi.de, kaempgen@fzi.de
[2] inovex GmbH
stefan.igel@inovex.de

Zusammenfassung. In den letzten Jahren ist die Menge der verfügbaren Linked Data im Web stetig gestiegen. Daher veröffentlichen immer mehr Provider ihre statistischen Datensätze als Linked Data, um sie mit weiteren Informationen anzureichern. Wir möchten in diesem Kurzbeitrag zu einer laufenden Arbeit eine Extract-Transform-Load (ETL) Pipeline vorstellen, die extrem große Mengen an Linked Data automatisiert in ein horizontal skalierbares Open Source OLAP-System bereitstellen kann.

Schlüsselwörter: Linked Data, Data Cube, Parallelisierung, MapReduce

1 Einleitung

In den letzten Jahren ist die Menge der verfügbaren Linked Data stetig gestiegen und immer mehr numerische Datensätze werden im Web mittels des RDF Data Cube Vokabulars (QB) als Linked Data veröffentlicht. Ein Vorteil besteht darin, die Bedeutung der numerischen Daten durch Verlinkung mit Zusatzinformationen näher zu bestimmen. Somit können beispielsweise Provenance-Informationen oder weitergehende Informationen (z.B. Anzahl der Mitarbeiter) hinzugefügt werden. Darüber hinaus können auch interne Daten mit den numerischen Daten verlinkt und zur Analyse verwendet werden.

Bevor Analysten jedoch in der Lage sind, Unternehmensleistungen vergleichen zu können, verbringen sie unverhältnismäßig viel Zeit mit der Identifizierung, Erfassung und Aufbereitung der relevanten Daten. Der Aufwand steigt mit der Anzahl der Datenquellen und damit unterschiedlichen Formaten oder Bezeichnungen für identische Objekte. Diese Prozesse müssen daher optimiert und möglichst automatisiert werden. Für entscheidungsunterstützende Analysen numerischer Datensätze bietet das Konzept OLAP (Online Analytical Processing) eine multidimensionale Betrachtung des Datenbestands.

 In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

In der Arbeit von Kämpgen und Harth [4] wurde ein Extract-, Transform- und Load-Prozess (ETL-Prozess) vorgestellt, der die statistischen Linked Data aus unterschiedlichen RDF Stores, unter Anwendung der Abfragesprache SPARQL und dem Cube-Vokabular QB, in ein multidimensionales Datenmodell transformiert. Ferner wurden die Informationen in diesem ETL-Prozess in einem relationalen Data Warehouse gespeichert. Auf diese Weise konnte mit der OLAP-to-SQL-Engine Mondrian [6] die Vorteile der multidimensionalen Abfragemöglichkeit und erweiterten Selektierbarkeit von OLAP-Anfragen mit MDX (Multidimensional Expressions) genutzt werden. Dieser Ansatz beinhaltet jedoch drei wesentliche Probleme:

(V1) Die Dauer des ETL-Prozesses bei großen Datensätzen mit vielen Zusatzinformationen ist nicht zufriedenstellend, da innerhalb der RDF-Daten die nötigen Informationen für das multidimensionale Datenmodell (Metadaten und Daten) herausgezogen werden müssen.
(V2) Bei einer Aktualisierung des Datenbestands oder bei neu hinzukommende Informationen muss der ETL-Prozess komplett neu durchgeführt werden.
(V3) Zusatzinformationen an den Datensätzen werden bei der Erstellung des multidimensionalen Datenmodells gefiltert und können bei analytischen Abfragen nicht berücksichtigt oder als Zusatzinformation abgefragt werden. Auch wenn das multidimensionale Datenmodell dementsprechend erweitert wird, können heterogene Zusatzinformationen nicht berücksichtigt werden.

In einer vorherigen Arbeiten [5] wurden die Daten in einem Triple-Store geladen, um analytische Abfragen mittels der graphenbasierten Sprache SPARQL auszuführen. Es zeigte sich, dass der Triple Store mit beliebigen RDF-Daten weniger effizient für analytische Abfragen geeignet ist als ein RDBMS mit Sternschema. Eine weitere Arbeit [3] beschäftigte sich mit der Optimierung eines Triple Stores durch horizontale Skalierung. Da NoSQL-Systeme für komplexe Operationen weniger geeignet sind, war die Ausführung der analytischen Abfragen nicht effizient genug. In der Arbeit von Abelló et al. [1] wurden analytische Abfragen auf MapReduce-basierten Systemen evaluiert. Dabei wurden die Vorteile von Big-Data-Technologien bei der Generierung eines OLAP Cubes für analytische Abfragen überprüft, jedoch ohne eine horizontale Skalierung durchzuführen. Grundlegend kann gesagt werden, dass diese drei Ansätze für die Analyse einer beliebigen Menge an RDF-Daten eine Herausforderung darstellen.

Bei der Analyse großer Datenmengen sind daher Big-Data-Technologien notwendig. Sowohl relationale Datenbanken, RDF Stores als auch OLAP Engines skalieren in der Regel nicht horizontal und besitzen daher eine natürliche Grenze bzgl. ihrer Datenspeicher- und Datenverarbeitungskapazität. Wir glauben, dass sich diese Beschränkungen mittels Parallelisierung über viele Rechner überwinden lassen. Mit Apache Hadoop sind derartige Technologien in einem Open Source Software Stack verfügbar. Bislang wurde nicht erforscht, ob eine enorme RDF-Datenmenge in einem automatisierten ETL-Prozess durch eine Umsetzung der Architektur von Kämpgen und Harth mit Hadoop-Komponenten für OLAP-Analysen bereitgestellt werden kann.

2 Aktueller Ansatz

Der hier präsentierte Lösungsansatz überführt Kämpgen und Harths Konzept in eine horizontal skalierende Architektur auf der Basis von Hadoop. Die nicht-skalierbaren Komponenten, wie die RDF-Datenbank, die Abfragesprache SPARQL und die relationale Datenbank, werden dabei durch Technologien und Frameworks aus dem Hadoop-Ökosystem ersetzt. Abbildung 1 veranschaulicht die neue Gesamtarchitektur.

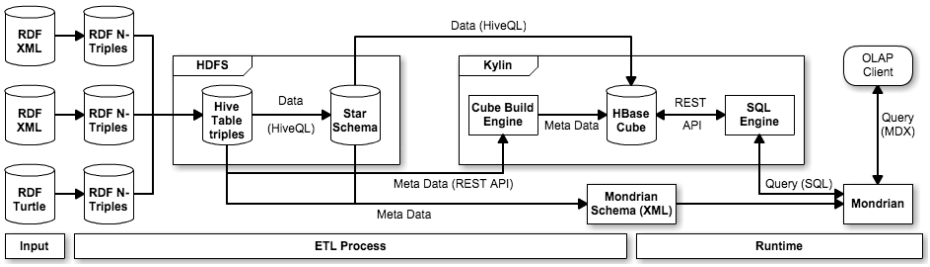

Abb. 1. Parallelisierungsarchitektur für analytische Abfragen auf Statistical Linked Data mit MapReduce-basiertem ETL-Prozess.

Die in verschiedenen RDF Stores abgelegten Linked Data werden in das N-Triples-Format umgewandelt und in das Hadoop Distributed File System (HDFS) geladen. Das zeilenbasierte N-Triples-Format ist besonders gut geeignet um die Daten in die Hive-Tabelle „triples“ mit den Spalten „subject“, „predicate“ und „object“ zu transformieren. Im Vergleich zur Arbeit von Cudré-Mauroux et al. [3] werden in unserem ETL-Prozess keine Property Tables generiert. Die vorkommenden Predicates werden beim Befüllen der Hive-Tabelle in Ordnern partitioniert. Dies optimiert Hive-Abfragen bei der Suche nach bestimmten Predicates, z. B. nach Measures (Predicate qb:measure). Unter Anwendung solcher Hive-Abfragen und MapReduce Jobs findet auf Basis der Definition des Cubes im QB-Vokabular eine Transformation der triples-Tabelle in ein relationales Datenmodell im Sternschema mit einer Fakten- und mehreren Dimensionstabellen statt. Die Cube Build Engine in Apache Kylin[2] erstellt aus dem Hive-Sternschema in mehreren MapReduce Jobs den OLAP Cube. Für die Speicherung der Cuboids ist die NoSQL-Datenbank HBase verantwortlich. In dieser spaltenorientierten NoSQL-Datenbank werden die verschiedenen Aggregationen der Cuboids gespeichert. HBase eignet sich, aufgrund der robusten Verarbeitung sehr großer Datenmengen und durch redundante, horizontale Verteilung durch das Ablegen der Daten ins HDFS, besonders gut als Speicherort der Cuboids.

Die SQL Engine in Kylin erlaubt das Absetzen von SQL-Anfragen an den Cube. Eine Ausführung von OLAP-Anfragen mit MDX ist bislang jedoch nicht möglich. Daher liegt ein weiteres Augenmerk dieser Arbeit in der Anpassung der OLAP-to-SQL Engine Mondrian. Im Mittelpunkt dieser Betrachtung steht

die Kommunikation zwischen dem OLAP Client und Kylin, besonders vor dem Hintergrund, dass Kylin lediglich eine ANSI-SQL-Teilmenge verarbeiten kann.

3 Vorläufige Evaluation

Grundlage der Evaluation ist die von Kämpgen und Hart vorgestellten Arbeit [5], die wiederum auf den Star Schema Benchmark (SSB) aufbaut, um analytische Abfragemethoden über Statistical Linked Data zu evaluieren. Die Generierung einer beliebigen Datenmenge im Sternschema wird durch das TPC-H sichergestellt. Dies erlaubt eine Untersuchung größerer RDF-Datenmengen im Hinblick auf die geplante Architektur. Zusätzlich stellt SSB unterschiedliche analytische SQL-Queries zur Verfügung, die eine detaillierte und vergleichbare Evaluierung der Architektur ermöglicht. In [5] wurden diese SQL-Queries aufgelistet, diskutiert und in vergleichbare MDX-Queries umgewandelt.

In dieser Arbeit werden mithilfe der TPC-H Benchmark-Datengenerierung verschieden große Datenmengen sowohl hinsichtlich der Ausführungsdauer des ETL-Prozesses als auch der Anfragedauer bei analytischen Queries untersucht. Für die erste Evaluation verwenden wir SSB mit der Skalierung 1 (ca. 6.000.000 Datensätze). Dies entspricht einem Umfang von 4,4GB an RDF-Daten im QB-Vokabular. Unser Cluster besteht aus drei virtuellen Rechnern mit Ubuntu 12.04 LTS und jeweils 32GB Ram, wobei jeder MapReduce Job max. 8GB zugewiesen bekommt. Der Hauptknoten hat dabei vier CPUs mit 2,5GHz, die restlichen Knoten besitzen jeweils zwei CPUs mit 2,4GHz.

Die Umwandlung der RDF-Daten in das zeilenbasierte N-Triples-Format dauert auf dem Hauptrechner durchschnittlich 1147s. Die resultierenden N-Triples-Dateien haben eine Gesamtgröße von 16,6GB und der Umzug ins HDFS dauert durchschnittlich 224s (75,92 MB/s). Der ETL-Prozess zur Bewirtschaftung des Sternschemas dauert durchschnittlich 5748s und die Generierung des OLAP-Cubes in Kylin benötigt auf diesem Cluster 2257s.

Bei unserer vorläufigen Evaluation findet, bis auf die Umwandlung der RDF-Daten in das N-Triples-Format, bereits an jeder möglichen Stelle eine Parallelisierung statt. Nach dem Umzug der Daten in das HDFS werden alle restlichen Schritte durch verschiedene MapReduce Jobs parallel ausgeführt. Die Speicherung des OLAP Cubes in HBase führt zu einer horizontalen Verteilung der Daten.

Aufgrund der Verwendung von Calculated Members und der Einschränkung der SQL-Syntax ist zum jetzigen Zeitpunkt eine Evaluation der MDX-Abfragen Q1, Q2 und Q3 in Kylin nicht möglich. Ein Lösungsansatz dieses Problems besteht darin, die Multiplikation der Measures zur Laufzeit des ETL-Prozesses zu berechnen und in die Faktentabelle als neue Spalte zu speichern.

Die durchschnittliche Ausführungsdauer der MDX-Abfragen ist in Tabelle 1 aufgelistet. Obwohl die vorläufige Evaluation auf einem Cluster mit virtuellen Rechnern ausgeführt wird, lässt sich erkennen, dass alle MDX-Abfragen nach dem ETL-Prozess mit Kylin schneller ausgeführt werden als bei einer traditionellen Datenbank wie MySQL. Eine systematische Evaluation in Abhängigkeit der Clustergröße sollte einen noch deutlicheren Unterschied bei der Ausführung

	Q1.1	Q1.2	Q1.3	Q2.1	Q2.2	Q2.3	Q3.1	Q3.2	Q3.3	Q3.4	Q4.1	Q4.2	Q4.3	Total
MySQL	42,4	40,9	41,7	16,2	17,3	15,2	12,4	10,6	9,6	7,3	15,4	10,7	10,7	250,4
Kylin	N/A	N/A	N/A	4,0	9,2	1,9	3,1	4,0	3,1	2,2	5,3	3,8	5,1	41,7

Tabelle 1. Ergebnisse der Prä-Evaluationen mit Ausführungsdauer pro MDX-Abfrage

der Abfragen aufzeigen. Ferner soll der ETL-Prozess und die Analysedauer bei größeren und mit Hintergrundinformationen angereicherten Datensätzen untersucht werden.

4 Zusammenfassung

Die vorgestellte, durchgängig horizontal skalierende Architektur auf Basis von Hadoop liefert eine vielversprechende Lösung für die effiziente Speicherung und performante Verarbeitung großer Linked Data Volumina mit Hadoop. Sie beinhaltet einen Ansatz zur skalierbaren Transformation dieser Daten mittels MapReduce und Hive. Aufsetzend auf Hive und HBase stellt Kylin multidimensionale Datenstrukturen zur Verfügung, die die Analyse großer Volumina numerischer Linked Data mittels etablierter OLAP-Methoden und Tools ermöglichen. Eine folgende systematische Evaluation bezüglich Skalierbarkeit und Performanz muss die ersten Ergebnisse allerdings noch bestätigen. Die beschriebene Architektur erscheint grundsätzlich auch geeignet zur effizienten Aktualisierung des Datenbestandes und zur Ergänzung der numerischen Daten um heterogene Zusatzinformationen. Dies wird Gegenstand zukünftiger Forschungsarbeiten sein.

Literatur

1. Abelló, A., Ferrarons, J., Romero, O.: Building Cubes with MapReduce. In: Proceedings of the ACM 14th international workshop on Data Warehousing and OLAP. pp. 17–24. ACM (2011)
2. Apache: Kylin – An Open Source Distributed Analytics Engine (2015), http://kylin.incubator.apache.org/, aufgerufen am 11. September 2015
3. Cudré-Mauroux, P., Enchev, I., Fundatureanu, S., Groth, P., Haque, A., Harth, A., Keppmann, F.L., Miranker, D., Sequeda, J.F., Wylot, M.: NoSQL databases for RDF: an empirical evaluation. In: The Semantic Web–ISWC 2013, pp. 310–325. Springer (2013)
4. Kämpgen, B., Harth, A.: Transforming Statistical Linked Data for Use in OLAP Systems. In: Proceedings of the 7th international conference on Semantic systems. pp. 33–40. ACM (2011)
5. Kämpgen, B., Harth, A.: No Size Fits All – Running the Star Schema Benchmark with SPARQL and RDF Aggregate Views. In: The Semantic Web: Semantics and Big Data, pp. 290–304. Springer (2013)
6. Pentaho: Mondrian - Open Source Business Analytics Engine (2015), http://community.pentaho.com/projects/mondrian/, aufgerufen am 11. September 2015