Research Seminar 1: Task 2 - Themenfindung

Markus Deutschl

Thema: Graphdatenbanken

Kurzbeschreibung

Eine Graphdatenbank (bzw. graphenorientierte Datenbank) verwendet zur Speicherung von Daten, wie der Name bereits sagt, Graphen mit Knoten und Kanten und gehört zu den gegenwärtig vier Modellen der NOSQL-Bewegung[1]. Dies eignet sich besonders gut zur Speicherung von stark vernetzten Daten. Diese Datenmodelle können in anderen Datenbanksystemen (DBS) nur bedingt abgebildet werden und sind bei Abfragen oft mit schlechter Laufzeit verbunden. Des Weiteren benötigen solche Datenstrukturen oft flexible Schemata, die relationale DBS nicht bieten können. Das Abfragemodell von Graphdatenbanken unterscheidet sich wesentlich von dem relationaler DBS, da hier eine Traversierung der Knoten und Kanten des Graphen vorgenommen wird. Die Verantwortlichkeit der Traversierung liegt hier nicht bei der Abfragesprache sondern bei der Programmlogik.

Da das Datenmodell sehr gut für Recommender Systems (RS) geeignet scheint, soll die Masterarbeit eine Analyse des Themengebiets beinhalten, sowie eine Evaluierung bestehender DBS in Bezug auf den Nutzen bzw. die Performance in Hinblick auf das für MovLib entwickelte RS einschließen.

Publikationen

Graph Databases

Diese Publikation beschäftigt sich damit, ein Graphdatenbanksystem zu implementieren. Es werden mehrere Ansätze präsentiert, wie ein solches designt und implementiert werden kann, sowie die Definition der Data Definition Language (DDL) und der Data Manipulation Language (DML) für die Systeme. Des Weiteren werden Konzepte vorgestellt, wie Updates durchgeführt und auch wie neue Konzepte eingeführt werden können, bzw. wie Queries höherer Ordnung umgesetzt werden können, die durch SQL nicht einfach verarbeitbar sind. Es wird auch gezeigt, wie bestehende relationale Datenbanken umgewandelt und die Daten wiederverwendet werden können. Die vorgestellten Systeme wurden jedoch nicht alle wirklich implementiert, aber jedenfalls alle konzeptuell vorgestellt. (Silvescu, Caragea and Atramenov 2002)

Survey of Graph Database Models

In dieser Abhandlung wird die gesamte Arbeit, die zum Verfassungszeitpunkt im Bereich der Graphdatenbanken präsent war, zusammengefasst. Im Speziellen wird auf die Punkte Graphdatenbankmodellierung, Datenstrukturen, Abfragesprachen und Integritätsconstraints eingegangen. (Angles and Gutierrez 2010)

[1] http://nosql-database.org/

A Comparison of a Graph Database and a Relational Database

Die Publikation zieht einen auf Messungen beruhenden Vergleich der beiden DBS MySQL und Neo4j. Dieser soll den Nutzen der getesteten Systeme in Bezug auf die Abfrage und die Speicherung von Herkunftsdaten evaluieren, die am besten durch Graphen abgebildet werden können und für die häufig Traversierungen nötig sind, um diese sinnvoll verarbeiten zu können. Um den Messvergleich anstellen zu können, wurden diese Daten in einem gerichteten azyklischen Graphen gespeichert und die Query-Laufzeiten von „Strukturabfragen“ und „Payload-Abfragen“ verglichen, welche nochmals in Unterkategorien aufgeteilt wurden. (Vicknair, et al. 2010)

The Graph Traversal Pattern

Die beiden Autoren geben zu Anfang eine kurze Einführung in die Graphentheorie und besprechen anschließend das „Graph Traversal Pattern“ und seinen Einsatz in der Verarbeitung von Daten. Hierbei ist die Rede vom Einsatz und Nutzen von Graphdatenbanken in der Praxis, für die auch Anwendungsbeispiele gegeben werden. Des Weiteren wird auf die Wichtigkeit der sinnvollen Graphpartitionierung aufgrund des Domänenmodells und die Indexierung von Graphen eingegangen, die in der Praxis erhebliche Performancesprünge bedeuten können. (Rodriguez and Neubauer 2011)

Graph Query Language: Implementing Graph Pattern Queries on a Relational Database

Der Kernpunkt dieser Arbeit konzentriert sich auf Graphdatenbanken, die auf relationalen Datenbanken aufsetzen. Dabei werden Graphabfragen in SQL übersetzt, was bei direkter Übersetzung in ein einzelnes SQL-Statement sehr schlechte Performance bedeutet. Es wird nach einer Einführung in Graphdatenbanken und deren Abfragen ein Ansatz vorgestellt, der mehrere SQL-Statements generiert, um zuerst in die Tiefe der sog. „Pattern Queries“ (Abfragen für Graphdatenbanken) vorzugehen und diese Resultate weiterzuverarbeiten. Dadurch wird zusätzliche Programmlogik in der Graph Query Engine benötigt. (Kaplan, et al. 2008)

HyperGraphDB: A Generalized Graph Database

In diesem Paper wird das DBS HyperGraphDB vorgestellt, das Hypergraphen zur Speicherung der Daten verwendet. Hierbei können sog. Hyperkanten andere Hyperkanten beinhalten und das Datenmodell wird somit weiter generalisiert. Dies impliziert, dass das System sehr gut an den Anwendungszweck angepasst werden kann, jedoch die Hauptrepräsentation der Daten einheitlich bleibt. Somit kann das Datenmodell auf den spezifischen Anwendungsfall optimiert werden. Das System selbst wurde als universelles Datenmodell für komplexe und große Wissensrepräsentationsanwendungen entwickelt, wie z.B. im Bereich der künstlichen Intelligenz oder der Bioinformatik. (Iordanov 2010)

Graph Database Filtering Using Decision Trees

Graphen sind generell ein sehr mächtiges Werkzeug, um strukturierte Daten abzubilden, haben jedoch eine hohe Verarbeitungskomplexität. Bei der Mustererkennung ist es oft erforderlich eine unbekannte Probe gegen eine Datenbank aus sog. Musterkandidaten zu vergleichen. Bei diesem Prozess spielt jedoch die Datenbankgröße als Faktor in die Komplexität eine zusätzliche Rolle. Die Autoren präsentieren aus diesem Grund einen Ansatz basierend auf Machine learning, um diesen Faktor zu reduzieren. Basierend auf den Eigenschaftsvektoren des Graphen erstellt die Methode einen Entscheidungsbaum, der die Datenbank indexiert. (Irniger and Bunke 2004)

Closure-Tree: An Index Structure for Graph Queries

Da Graphenabfragen immer mehr an Bedeutung gewinnen, ist auch die Indexierung dieser sehr wichtig. Diese macht Abfragen wesentlich effizienter und senkt somit die Laufzeit dieser. Die beiden Autoren präsentieren eine Methode, die Closure-Tree genannt wird und Graphen hierarchisch organisiert. Der Kernpunkt ist, dass jeder Knoten seine Nachfolger durch eine Graph Closure zusammenfasst. Die vorgestellte Technik unterstützt sowohl Subgraph- wie auch Ähnlichkeitsabfragen. Für Subgraph-Queries wird hier eine Methode verwendet, die den Isomorphismus von Subgraphen mit hoher Genauigkeit nähern kann. Die Ähnlichkeitsabfragen werden durch Bearbeitungsdistanzen mittels heuristischen Graph-Abbildungsmethoden optimiert. (He and Singh 2006)

A Graph Model for E-Commerce Recommender Systems

Dieses Paper behandelt die Entwicklung eines Graphenmodells, welches eine generische Datenrepräsentation zur Verfügung stellt und verschiedene Empfehlungsmethoden unterstützt. Um die Nützlichkeit zu zeigen wurden die drei Empfehlungsmethoden „direct retrieval“, „association mining“ und „high-degree association retrieval“ implementiert, die gegen das Datenset eines Online-Buchgeschäfts evaluiert wurden. Die Messungen haben ergeben, dass die Kombination aus Produktinformationen mit der Transaktionsgeschichte des Kunden genauere Vorhersagen lieferte als die reine Verwendung von kollaborativen Daten. (Huang, Chung and Chen 2003)

Movie Recommendation using Random Walks over the Contextual Graph

Die meisten Recommender Systems verwenden zur Generierung der Empfehlungen die Beziehungen von Benutzern zu Items und/oder die Bewertungen. Dies bezieht den Benutzerkontext (z.B. demographische Daten) bzw. den Item-Kontext (bei Filmen z.B. Schauspielerbeziehungen) nicht ein, was jedoch zu besseren Ergebnissen führen kann. In dieser Arbeit wird deshalb ein Empfehlungsalgorithmus namens „ContextWalk“ vorgestellt, der Empfehlungen durch Random Walks über den Kontextgraphen der Daten generiert. Dieser Algorithmus ist speziell für Filmempfehlungen entwickelt worden. Leider werden keine Messungen zur Verifikation der Nützlichkeit des Algorithmus angegeben. (Bogers 2010)

Literaturverzeichnis

Angles, Renzo, und Claudio Gutierrez. „Survey of Graph Database Models.“ *Technical Report TR/DCC-2005-10.* Oktober 2010.

Bogers, Toine. *Movie Recommendation using Random Walks over the Contextual Graph.* Kopenhagen, 2010.

He, Huahai, und Ambuj K. Singh. *Closure-Tree: An Index Structure for Graph Queries.* Herausgeber: IEEE. Santa Barbara, 2006.

Huang, Zan, Wingyan Chung, und Hsinchun Chen. *A Graph Model for E-Commerce Recommender Systems.* Herausgeber: Wiley Periodics. Tucson, 13. August 2003.

Iordanov, Boris. *HyperGraphDB: A Generalized Graph Database.* 2010.

Irniger, Chritophe, und Horst Bunke. *Graph Database Filtering Using Decision Trees.* Herausgeber: IEEE. Bern, 2004.

Kaplan, Ian L., Ghaleb M. Abdulla, S. Terry Brugger, und Scott R. Kohn. „Implementing Graph Pattern Queries on a Relational Database.“ *LLNL technical report LLNL-TR-400310.* Livermore, 8. Januar 2008.

Rodriguez, Marko A., und Peter Neubauer. *The Graph Traversal Pattern.* 2011.

Silvescu, Adrian, Doina Caragea, und Anna Atramenov. *Graph Databases.* Herausgeber: Iowa State University. Ames, Iowa, Mai 2002.

Vicknair, Chad, Michael Macias, Zhendong Zhao, Xiaofei Nan, Yixin Chen, und Dawn Wilkins. *A Comparison of a Graph Database and a Relational Database.* Herausgeber: ACM. Oxford, April 2010.