Peer-Daten-Management-Systeme – PDMS*

Felix Naumann
Hasso-Plattner-Institut, Potsdam
naumann@hpi.uni-potsdam.de

Armin Roth
Hasso-Plattner-Institut, Potsdam
Armin.Roth@hpi.uni-potsdam.de

1 PDMS Architektur

Peer Daten Management Systeme (PDMS) sind die natürliche Erweiterung föderierter Informationssysteme. An Stelle einer zentralen Komponente, die ein integriertes Schema hält, treten einzelne Peers, die sowohl die Rolle von Datenquellen als auch die Rolle einer integrierten Komponente annehmen. Der Aufbau eines PDMS ist geprägt durch die Erstellung von Schema Mappings zwischen den Schemata einzelner Peers. Nur sie erlauben die Transformation von Anfragen eines Peers zu Anfragen an benachbarte Peers. Anfragen dürfen an jedes und von jedem an der Integration beteiligten System gestellt werden. Dieses wird dann versuchen, mittels der eigenen und anderer Daten Antworten zu berechnen. In Anlehnung an die Ideen von Peer-to-Peer-Systemen (P2P), insbesondere nämlich die Aufgabe der hierarchischen Architektur zugunsten einer netzartigen Struktur und der Verzicht auf globale Kontrolle, nennt man die resultierende Architektur *Peer-Daten-Management-Systeme*. Abbildung 1 zeigt ein kleines Netzwerk aus Peers, jeweils mit eigenem Schema und Mappings zwischen ihnen, und teilweise mit eigenen Datenbanken.

2 Die Rollen der Peers

Ein Peer in einem PDMS erfüllt zugleich die Rolle einer Datenquelle und die Rolle eines Mediators. Peers stellen ein Schema zur Verfügung, speichern Daten gemäß diesem Schema, nehmen Anfragen entgegen und reichen Anfrageergebnisse zurück. In ihrer *Rolle als Datenquelle* kommen die Antworten aus eigenen, lokalen Datenbeständen (z.B. Peer 1 in Abbildung 1). In ihrer *Rolle als Mediator* benutzen sie andere Peers, um Antworten zu finden. Nach außen, also für die anderen Peers des PDMS, ist es kein Unterschied, wo die Daten nun tatsächlich herkommen.

Als Mediator speichern Peers Korrespondenzen oder Schema Mappings zwischen dem eigenen Schema und den Schemata ausgewählter anderer Peers. Mappings beschreiben äquivalente Elemente zwischen zwei Schemata und spezifizieren, wie Daten transformiert werden müssen, wenn sie von einem Peer zum anderen übertragen werden. Sie übernehmen damit die Rolle von Wrappern in mediatorbasierten Systemen. Betrachtet man Peers als Knoten und Mappings als Kanten, bilden die Peers eines PDMS ein *Netzwerk von Datenquellen* (siehe Abbildung 1). Anfragen an einen Peer werden sowohl mit eigenen Daten beantwortet (sofern vorhanden) als auch über die durch Korrespondenzen verbundenen Peers.

3 Anfrageplanung in PDMS

Die Schemata einzelner Peers sind mit Korrespondenzen untereinander verbunden, die oft Local-as-View- oder Global-as-View-Form haben. Einen anderen Ansatz verfolgt das PDMS Hyperion [1], in dem Korrespondenzen zwischen Tupeln unterschiedlicher Peers in so genannten *mapping tables* verwaltet werden.

Nutzeranfragen werden an einen einzelnen Peer gestellt. Dieser speichert kann gegebenenfalls Teile des Ergebnisses liefern, muss die Anfrage aber auch an *geeignete benachbarte Peers versenden*, also an Peers, die über eine Korrespondenz mit für die Anfrage relevanten Relationen verknüpft sind. Aufgrund der Heterogenität zwischen Peers kann die

*Der vorliegende Text ist aus [5] adaptiert.

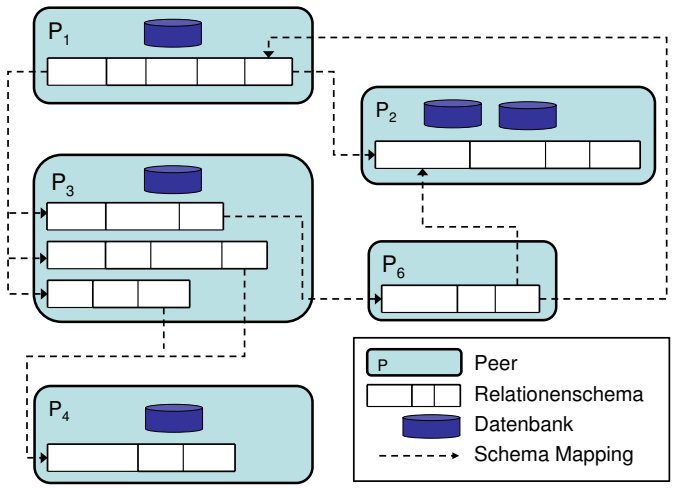

Abbildung 1: Ein Peer-Daten-Management-System

ursprüngliche Anfrage nicht direkt weitergeleitet werden, sondern muss zuvor gemäß der LaV- bzw. GaV-Regeln umgeschrieben werden [7].

Die benachbarten Peers können selbst zum Anfrageergebnis beisteuern, geben aber die Anfrage auch an ihre Nachbarn weiter. Im relationalen PDMS Piazza entsteht so ein Anfragebaum, der so genannte *rule-goal-tree* [3]. Zur Anfrageplanung werden die Blätter des Baumes immer weiter entfaltet, bis sie einzig aus lokalen Relationen bestehen. Der rule-goal-tree wird nun in einen ausführbaren Anfrageplan umgeschrieben, der weiter optimiert werden kann. In [4] wird ein entsprechender Algorithmus für XML Daten beschrieben.

Ein typisches Merkmal solcher Pläne ist deren *hohe Redundanz*: Eine Relation eines einzelnen Peers kann in einem Anfrageplan sehr oft in unterschiedlichen „Rollen“ auftauchen. Dadurch werden gleiche Anfrageergebnisse mehrfach erzeugt und müssen im Endergebnis wieder entfernt werden. Im Piazza System ist eine Methode umgesetzt, die Teilbäume entfernt, von denen sicher ist, dass die durch sie berechneten Ergebnisse bereits in einem anderen Teilbaum berechnet werden. In [6] wird zusätzlich beschrieben, wie man gezielt Teilbäume entfernen kann, die nur wenige Daten zum Gesamtergebnis beisteuern.

4 Vor- und Nachteile der PDMS-Architektur

PDMS befinden sich zwischen zwei Extremen: *Föderierte DBMS* (FDBMS) mit einer festen Menge an Datenquellen und festen Zuordnungen zwischen einem globalen und mehreren lokalen Schemata auf der einen Seite [2] und den hochdynamischen *P2P-Systemen* mit ständig wechselnden Beteiligten auf der anderen Seite. In den folgenden Abschnitten vergleichen wir den PDMS-Ansatz mit beiden Extremen.

Die Vorteile von FDBMS, wie etwa Ortstransparenz und Schematransparenz, gelten auch für PDMS. Der wesentliche Vorteil von PDMS gegenüber FDBMS ist deren Flexibilität. Um eine neue Datenquelle hinzuzufügen, reicht es in der Regel, ein einziges Schema Mapping zu definieren, und der Peer ist in das Netzwerk des PDMS aufgenommen. Da sämtliche Quellen über Mappings miteinander verbunden sind, kann das Mapping von der neuen Datenquelle zu der ähnlichsten bereits vorhandenen Datenquelle abbilden. Somit fällt die Definition vergleichsweise leicht. Zudem beeinträchtigt das Hinzufügen und Entfernen einzelner Datenquellen das Gesamtsystem nur wenig: Es gibt kein globales Schema, das zu ändern wäre. Lediglich Mappings müssen erzeugt bzw. entfernt werden, und diese Mappings werden nur lokal von den Peers verwaltet.

Ein wesentlicher Nachteil von PDMS ist die *komplexe Anfragebearbeitung über viele Peers* hinweg. Anfragen

werden vielfach umformuliert, während sie von einem Peer zum nächsten gereicht werden. Dabei müssen Zyklen vermieden werden. Umgekehrt werden Daten auf dem Rückweg durch die Peers vielfach transformiert, was unter anderem einen schleichenden Verlust an Semantik und Qualität bedeuten kann.

Während einem FDBMS alle Datenquellen bekannt sind, die Antworten zu einer Anfrage beisteuern, ist dies in einem PDMS nicht der Fall. Der Peer, an dem die Anfrage gestellt wird, kennt zwar seine direkten Nachbarn und leitet gegebenenfalls die Anfrage an diese Peers weiter. Diese können jedoch wiederum weitere Peers im Netzwerk befragen, um schließlich eine gebündelte Antwort an den ursprünglichen Peer zurückzusenden. Ohne aufwändigen Austausch von Metadaten weiß dieser nicht, welche Peers letztlich am Ergebnis beteiligt waren. Selbst wenn diese oft komplexen Informationen zusammen mit den Antworten geliefert werden, ist es dem anfragenden Peer im Gegensatz zum FDBMS nicht schon zum Anfragezeitpunkt möglich festzulegen, welche Peers Antworten liefern dürfen bzw. sollen.

PDMS sind als Erweiterung von FDBMS um P2P-Techniken entstanden. Es bestehen jedoch wesentliche Unterschiede zu beiden Architekturen, sowohl auf der logischen Ebene (Schemata, Anfragen, Anfragebearbeitung) als auch in ihrer physischen Ausprägung (Dynamik und Größe). Tabelle 1 fasst diese zusammen.

5 Ausblick

In dynamischen Systemen wie PDMS können Aussagen über Daten anderer Peers (z.B. Mappings, Vollständigkeit) manchmal nicht mit Sicherheit, sondern nur mit einer gewissen Wahrscheinlichkeit oder auf statistischer Basis getroffen werden. Diese Wahrscheinlichkeiten und Statistiken aufzubauen und in einer approximativen Anfragebearbeitung zu nutzen, sind vielversprechende Forschungsrichtungen. Auch zur Berücksichtigung der Datenqualität in PDMS sind solche Ansätze sinnvoll.

Dezentral entstandene Netzwerke von Schema Mappings zwischen den Peers bieten häufig Verbesserungspotential. So lässt sich Informationsverlust entlang von Mapping-Pfaden verringern, indem lange Pfade durch semantisch weniger verlustreichere ersetzt werden. Ein solches Management von Mapping-Netzwerken erfordert aber zu einem gewissen Grad globale Koordination zwischen einer Gruppe von Peers.

Aspekte der Informationssicherheit spielen in Forschungsansätzen zu PDMS bisher noch eine untergeordnete Rolle. Wenn Daten jedoch über mehrere Peers weitergereicht werden, kommt der Zugriffskontrolle eine wichtige Bedeutung zu.

Zusammenfassend kann beobachtet werden, dass der Datenaustausch innerhalb und zwischen Organisationen in der Praxis meist dezentral, also entsprechend dem Peer-to-Peer-Paradigma organisiert ist. Beispiele sind das Supply-Chain Management, das Management wissenschaftlicher Daten, Krankenhausinformationssysteme etc. Aufgrund des nicht generell notwendigen globalen Schemas ist dieser Ansatz flexibler und besser skalierbar. Daher könnten PDMS künftig die Entwicklung anderer Architekturen verteilter Informationssysteme, wie z.B. service-orientierte Ansätze beeinflussen.

Literatur

[1] Marcelo Arenas, Vasiliki Kantere, Anastasios Kementsietsidis, Iluju Kiringa, Renée J. Miller, and John Mylopoulos. The Hyperion project: from data integration to data coordination. *SIGMOD Record*, 32(3):53–58, 2003.

[2] Stefan Conrad. *Föderierte Datenbanksysteme: Konzepte der Datenintegration.* Springer Verlag, Berlin – Heidelberg – New York, September 1997.

[3] Alon Y. Halevy, Zachary Ives, Dan Suciu, and Igor Tatarinov. Schema mediation in peer data management systems. In *Proceedings of the International Conference on Data Engineering (ICDE)*, 2003.

[4] Alon Y. Halevy, Zachary G. Ives, Peter Mork, and Igor Tatarinov. Piazza: data management infrastructure for semantic web applications. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 556–567, 2003.

[5] Ulf Leser and Felix Naumann. *Informationsintegration.* dpunkt Verlag, Heidelberg, 2006.

	P2P	**PDMS**	**FDBMS**
Granularität	Dateien	Objekte, Tupel, Attribute	Objekte, Tupel, Attribute
Anfragen	Einfach: Suche	Komplex: SQL, XQuery etc.	Komplex: SQL, XQuery etc.
Wissen um andere Peers	Kein globales Wissen	Kenntnis von Nachbarn	Kenntnis aller Datenquellen
Anfragebearbeitung	Fluten des Netzes, verteilte Hashtabellen etc.	Gezieltes Verfolgen geeigneter Mappings	Direkte Anfragen an alle relevanten Quellen
Schema	Kein Schema (bzw. nur eine Relation)	Komplexes Schema	Komplexes Schema
Anzahl Peers	Hunderttausende	Unter hundert	Unter zehn
Dynamik	Hoch: Kurze Aufenthalte, spontanes Verlassen	Kontrolliert: Lange Aufenthalte, An- und Abmeldung	Kaum: Stabile Datenquellen

Tabelle 1: Vergleich zwischen P2P-Dateitausch und PDMS

[6] Armin Roth and Felix Naumann. Benefit and cost of query answering in PDMS. In *Proceedings of the International Workshop on Databases, Information Systems and Peer-to-Peer Computing (DBISP2P)*, 2005.

[7] Jeffrey D. Ullman. Information integration using logical views. In *Proceedings of the International Conference on Database Theory (ICDT)*, pages 19–40, Delphi, Greece, 1997.