Dynamische Datenintegration in Grid-Umgebungen

Stefan Conrad

Fachbereich Informatik
Universität Hamburg

0conrad@informatik.uni-hamburg.de

Zusammenfassung

Im Bereich des verteilten Rechnens in Grid-Umgebungen gibt es verschiedene Standardlösungen. Die Einbindung von Datenbanken in Computer-Grids geschieht bisher meist manuell und auf kleine Benutzergruppen beschränkt. Das DynaGrid-Projekt an der Universität Hamburg ermöglicht den Umgang mit hochdynamischem Einfügen und Entfernen von Datenquellen in Grid-Umgebungen sowie Ad-hoc-Anfragen an das System. Ziel der im Folgenden vorgestellten Diplomarbeit ist der Entwurf einer zentralen Komponente im DynaGrid, die Benutzeranfragen an zum Anfragezeitpunkt unbekannte Datenquellen entgegen nimmt, auf passende Datenquellen verteilt und deren Teilergebnisse integriert.

1 Grid-Computing

Die Open-Grid-Services-Architecture der Globus-Alliance [3] baut auf bestehenden Web-Service-Standards auf und schafft eine Infrastruktur für Grid-Services. Auf diesen Standard aufbauend hat die Arbeitsgruppe „Data Access and Integration Services“ (DAIS) des Global Grid Forum (GGF) eine Architektur entwickelt, um Datenbank-Zugriff im Grid zu ermöglichen. Open Grid Services Architecture – Data Access and Integration (OGSA-DAI) [4] ist eine Referenzimplementierung dieser Spezifikation. OGSA-DAI sieht drei zentrale Komponenten vor: die „DAIS-Group-Registry“ (DAISGR), die „Grid-Data-Service-Factoriy“ (GDSF) und als letztes den Grid-Data-Service (GDS). Die DAISGR bildet den Einstiegspunkt ins Grid. Sie ist eine Registratur für GDSF. Über eine solche Fabrik kann ein Nutzer eine Instanz eines Grid-Data-Service erhalten. Die Interaktion mit der eigentlichen Datenquelle findet über diesen Grid-Data-Service statt, indem ein XML-Dokument mit der Anfrage (Perform-Dokument) an den GDS geschickt wird. Das Ergebnis der Anfrage wird in einem neuen XML-Dokument (Response-Dokument) an den Nutzer zurückgeschickt.

Das OGSA-DAI-Rahmenwerk bietet Möglichkeiten, Datenbanken in Grid-Umgebungen einzusetzen; bisher ist dieser Einsatz jedoch sehr statisch. Es können die bei der Registry angemeldeten Datenquellen abgefragt werden, doch die Auswahl, welche die passende ist, bleibt dem Nutzer überlassen. Er muss sich aus eine Flut von Schemainformationen, die er von der Factory bekommt, die für die Anfrage benötigten Teile heraussuchen. Sollen mehrere Datenquellen befragt werden, muss dies für jede Datenquelle wiederholt werden.

Das DynaGrid-Projekt hat zum Ziel, die Abläufe innerhalb des OGSA-DAI-Rahmenwerkes zu automatisieren und zu dynamisieren. Dafür wurde zunächst die Registry erweitert [1]. Sie verwaltet nun zusätzliche Metadaten der Datenquellen, um zu entscheiden, welche Datenquellen potentiell zu einer Anfrage passen. Die Chance, eine passende Datenquelle zu finden, wird erhöht, indem die erweiterte Registry die Anfrage auch an andere ihr bekannte Registries weiterleitet und die Ergebnisse zusammenfasst.

Die Kommunikation zwischen Nutzer und Grid-Komponenten wird durch eine neue Komponente, die „Query-Engine“, gekapselt (siehe auch Abb. 1). Jegliche grid-interne Kommunikation

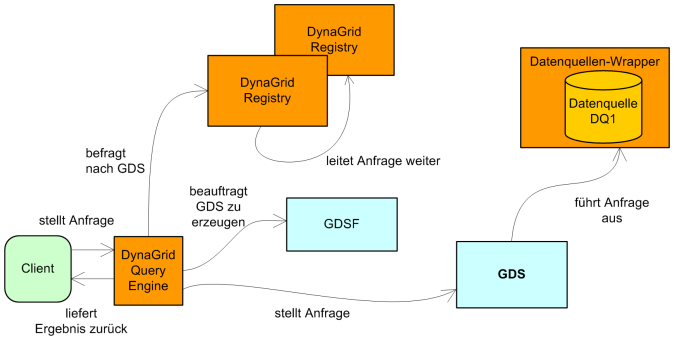

Abbildung 1: OGSA-DAI-Architektur durch DynaGrid-Komponenten (eingefärbt) erweitert

wird vor dem Nutzer verborgen – es wird nur das Gesamtergebnis ausgegeben. Die Query-Engine wird im Folgenden genauer betrachtet werden.

Die Datenquellen werden in der DynaGrid-Architektur nicht direkt angesprochen, sondern sind durch Wrapper gekapselt. Die Wrapper erzeugen Metadaten für die Registry, bieten eine einheitliche Schnittstelle und setzen die strukturunabhängigen Anfragen (s. Abschnitt 2) in der jeweiligen Datenquelle um.

2 Unbekannte Datenstrukturen

Zum Zeitpunkt der Nutzeranfrage ist unbekannt, welche Datenquellen zur Verfügung stehen. Somit kann die Anfrage nicht, wie es beispielsweise bei SQL (Structured Query Language) der Fall ist, gegen eine bekannte Struktur gestellt werden. Eine genaue Spezifikation, an welcher Stelle sich Informationen befinden, wie in SQL ist ohne Kenntnis der Struktur nicht möglich. Dieses Problem ließe sich durch eine Voranfrage und anschließende Anpassung der Anfrage an die erhaltene Struktur umgehen. Dieser Ansatz ist aufwendig und der Aufwand steigt proportional mit der Anzahl befragter Datenquellen. Daher wird im DynaGrid-Ansatz von der Struktur der Datenquellen abstrahiert. Der Nutzer beschreibt die gesuchte Information mit Anfrage-Entitäten, die benannt und durch Qualifikatoren eingeschränkt werden. Eine solche Entität stellt eine übergeordnete Struktur für die gesuchte Information dar. Der Name der Anfrage-Entität entspricht in etwa einem Tabellennamen einer relationalen Datenbank oder einem Elementnamen eines XML-Dokuments. Die Qualifikatoren, die einer Anfrage-Entität auferlegt werden können, sind Tripel aus Eigenschaft der Entität, Vergleichsoperator und Wert der Eigenschaft. Ihre Funktion entspricht der WHERE-Klausel in SQL oder XQuery. Durch die Angabe mehrerer Entitäten pro Anfrage können komplexe Anfragen formuliert werden, da dieses in etwa einer Erweiterung der SQL-FROM-Klausel entspricht. Es werden dann zusammenhängende Entitäten gesucht, beispielsweise Tabellen, die über eine Fremdschlüsselbeziehung verbunden sind. Vertiefende Informationen zur Verarbeitung von strukturlosen Anfragen in Datenquellen finden sich in [2].

3 Heterogenität

Die Heterogenität der verschiedenen Datenquellen wird durch die bereits genannten Wrapper gemindert, da die Wrapper eine einheitliche Schnittstelle bieten, unabhängig davon, ob es sich um eine relationale Datenbank oder ein XML-Repository handelt, und unabhängig von konkreten Herstellern und Produkten. Unterschiede in der Datenmodellierung und den Datenstrukturen werden ebenfalls durch die Wrapper ausgeglichen, indem die Anfrage strukturell an die entsprechenden Daten in der Datenquelle angepasst wird.

Die unterschiedliche Benennung von Relationen, Attributen und Elementen wird durch eine interne Normalisierung der benutzten Begriffe beseitigt. Hierfür werden sowohl bei der Anfrage

als auch bei den Resultaten der Datenquellen Bezeichnungen bei einem Ontologie-Service eingereicht und diese Begriffe ggf. durch einheitliche Begriffe ersetzt. Dieser Ontologie-Service kann zur Vereinfachung auf eingegrenzte Themengebiete eingeschränkt sein, um bessere Resultate zu liefern. Auch ist es möglich, einen solchen Service durch Experten erstellen und bei Erstanmeldung einer Datenquelle überarbeiten zu lassen, solange es keine praktikablen Lösungsansätze zur Erstellung von Ontologien gibt.

4 Datenintegration

Eine Anfrage wird parallel an mehrere Datenquellen geschickt, die von der Registry als geeignet eingestuft wurden. Die Ergebnisse sind minimale Ausschnitte aus den Datenquellen, die alle in der jeweiligen Datenquelle vorhandenen Anfrage-Entitäten enthalten. Die Response-Dokumente enthalten neben den Ergebnisdaten auch Metadaten über deren Struktur. Die Ergebnisse haben eine bekannte, festgelegte Struktur und durch die oben genannte Normalisierung eindeutige Namen, wodurch die Schemaintegration vereinfacht wird.

Aus der Anfrage wird zunächst ein vorläufiges erwartetes Antwortschema erzeugt. Mittels der Antwortstruktur-Metadaten der Teilergebnisse wird das erwartete Antwortschema gegebenenfalls ergänzt, wenn die Ergebnisse umfassender sind als erwartet. Diese Ergänzung findet für jedes Teilergebnis statt und so wird iterativ das partielle globale Schema erzeugt.

Nach der Schemaintegration wird das globale Schema mit den Daten aus den Teilergebnissen gefüllt und an den Nutzer ausgegeben.

5 Fazit

Ziel des DynaGrid-Ansatzes ist es, die relativ starren Ansätze des OGSA-DAI-Projektes aufzubrechen. In einer Grid-Umgebung muss es möglich sein, zur Laufzeit die zu einer Anfrage passenden, vorhandenen Datenquellen zu bestimmen und die Anfrage entsprechend zu stellen. Um das zu erreichen, wird die Anfrage von den Strukturen der Datenquellen entkoppelt, werden die passenden Datenquellen automatisch von der Registry zurückgegeben und stellen die Datenquellen eine einheitliche Schnittstelle zur Verfügung. Die Integration von Anfrageergebnissen geschieht ebenfalls dynamisch und benutzerunabhängig. Eine dynamisch erzeugte Antwortstruktur und eine interne Normalisierung der Begriffe schaffen die Voraussetzungen dafür.

Die Qualität einer vollautomatisierten Integration kann nicht an die Ergebnisse einer Integration mit manuellen Eingriffen heranreichen. Mit dem DynaGrid-Ansatz lassen sich Integrationsprobleme auf der strukturellen Ebene bewältigen. Für den Umgang mit Integrationskonflikten bei den konkreten Ausprägungen der Daten gibt es noch kein endgültiges Verfahren. Ob sich diese Konflikte ebenfalls durch die Verwendung des Ontologie-Service oder durch andere Verfahren lösen lassen, ist Bestandteil laufender Forschung.

Literatur

[1] Dreyer, Christian. *Entwurf und Realisierung eines erweiterten Verzeichnisdienstes für das Grid Data Computing* Diplomarbeit. Universität Hamburg, 2004

[2] Grohmann, René. *Wrapper-basierte Integration heterogener Datenquellen im Grid Data Computing* Diplomarbeit. Universität Hamburg, 2005

[3] The Globus Alliance. *Open Grid Services Architecture* http://www.globus.org/ogsa/, 2004

[4] UK Database Task Force *Open Grid Services Architecture Data Access and Integration* http://www.ogsadai.org/, 2003