Update-Propagation in gestaffelten und verteilten Caches

Marko Vrhovnik

Universität Stuttgart, Fakultät Informatik, Elektrotechnik und Informationstechnik
Institut für Parallele und Verteilte Systeme (IPVS), Abteilung Anwendersoftware (AS),
Universitätsstrasse 38, D-70569 Stuttgart
Marko.Vrhovnik@studi.informatik.uni-stuttgart.de

Einleitung

In der Forschungsgruppe NEXUS der Universität Stuttgart [5] wird eine offene Plattform für ortsbezogene Anwendungen entwickelt. Durch die Offenheit können beliebige Datenanbieter ihre Informationen durch die NEXUS-Plattform bereitstellen. Eine Föderations-Middleware verbirgt vor einer Anwendung die Verteilung der Daten und kombiniert Daten verschiedener Anbieter in geeigneter Weise. Zur Beschleunigung des Datenzugriffs werden in der NEXUS-Plattform Zwischenspeicher (Caches) eingesetzt, u.a. in der Föderations-Middleware, als auch in mobilen Endgeräten, auf denen NEXUS-Anwendungen typischerweise ausgeführt werden. Durch eine solche Staffelung und Verteilung von Caches können Daten „näher“ an eine Anwendung positioniert und entsprechend schneller geliefert werden. Um die Konsistenz zwischengespeicherter Daten sicherstellen zu können, müssen Aktualisierungen auch in den Caches vollzogen werden. In dieser Diplomarbeit wurde der Lösungsraum zur Propagation von Aktualisierungen (Updates) zu den jeweiligen Caches durchleuchtet. Dabei wurden verschiedene Cache-Konsistenzsemantiken erstellt und Strategien entwickelt, wie diese im NEXUS-System umgesetzt werden können. Ferner wurde untersucht, welche Auswirkungen die einzelnen Lösungsansätze auf die Autonomie der einzelnen Komponenten der NEXUS-Plattform haben.

Die Architektur der NEXUS-Plattform

Die NEXUS-Plattform besteht aus drei Ebenen (Abbildung 1): der Anwendungs-, der Föderations- und der Dienstebene.

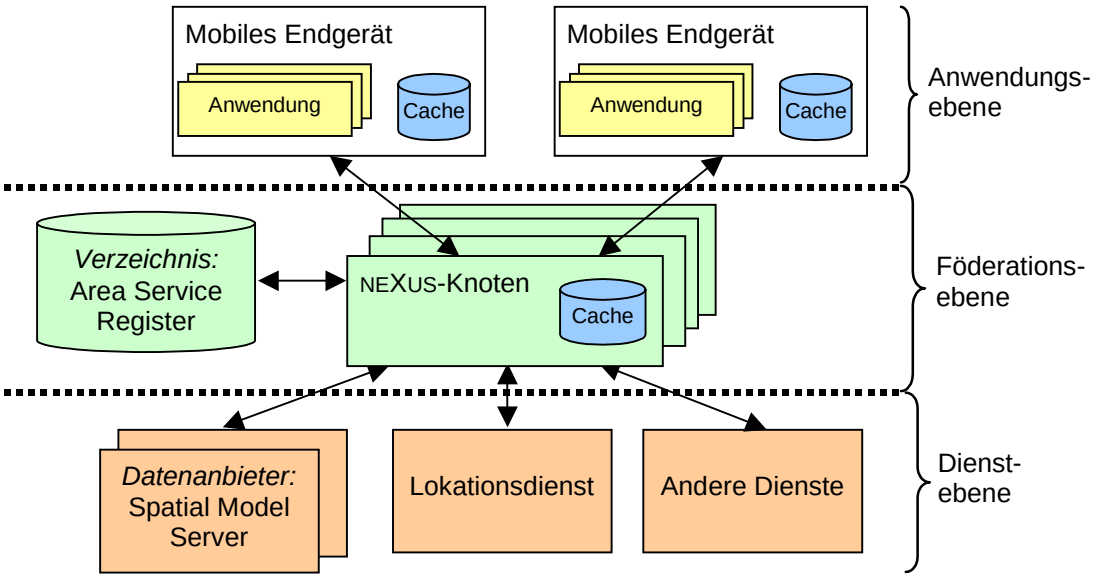

Abbildung 1: *Architektur der NEXUS-Plattform*

Die Föderationsebene wird aus mehreren unabhängigen NEXUS-Knoten (Föderationsknoten) gebildet. Ein *Föderationsknoten* nimmt Anfragen von Anwendungen entgegen, die typischerweise auf einem *mobilen Endgerät* (mobilen Client) ausgeführt werden und verteilt diese an die entsprechenden Komponenten der Dienstebene. Dazu wird das *Area Service Register* benötigt, ein Verzeichnis, das die

Adressen der Datenanbieter, das von ihren Daten abgedeckte Gebiet und die von ihnen gespeicherten Datentypen enthält. Damit kann ein Föderationsknoten herausfinden, welche Datenanbieter für eine bestimmte Anfrage zuständig sind. Der Föderationsknoten leitet die Anfrage an diese weiter und fasst das Ergebnis für die Anwendungen zusammen. Bei den Komponenten der Dienstebene handelt es sich um autonome Server, die von verschiedenen Anbietern aufgestellt und über das Area Service Register der NEXUS-Plattform bekannt gemacht werden. Daten über statische Objekte wie Gebäude, Straßen oder Verweise auf ortsrelevante Webseiten werden von sog. *Spatial Model Servern* gespeichert. Für mobile Objekte, die ihre Position häufig ändern, steht der *Lokationsdienst* zur Verfügung. Ferner verfügt die NEXUS-Plattform über weitere Dienste, die hier nicht näher betrachtet werden.

Das Caching-System der NEXUS-Plattform

Sowohl ein Föderationsknoten als auch ein mobiler Client kann einen Cache besitzen, in dem Ergebnismengen von Anfragen temporär zwischengespeichert werden, um künftige Datenanforderungen potentiell schneller bedienen zu können. Der Cache-Inhalt eines Föderationsknotens umfasst dabei Daten aller Anfragen, die über ihn abgewickelt werden, wohingegen der Cache-Inhalt eines mobilen Clients auf lokale Anfrageergebnisse der Anwendungen, die auf ihm ausgeführt werden, beschränkt bleibt. Im Unterschied zu einem Web-Cache werden im NEXUS-System nicht einzelne Objekte, sondern Ergebnismengen referenziert, die ein bestimmtes Anfrageprädikat erfüllen. Deshalb müssen Cache-Inhalte semantisch mit Hilfe von *Prädikaten* (Cache-Descriptions) beschrieben werden, damit bei einer vorliegenden Anfrage entschieden werden kann, ob eine angeforderte Datenmenge im Cache verfügbar ist, oder neu geladen werden muss.
Um die Konsistenz einer zwischengespeicherten Ergebnismenge gewährleisten zu können, müssen Datenänderungen in allen betroffenen Caches nachgezogen werden, wenn die Ursprungsdaten in der Dienstebene aktualisiert werden. Dabei muss berücksichtigt werden, dass mobile Clients lediglich über eine schwache, unzuverlässige Netzwerkverbindung verfügen, die zudem mit anderen mobilen Clients geteilt werden muss. Ferner ist zu beachten, dass Aktualisierungen an eine potentiell beliebig hohe Anzahl mobiler Clients propagiert werden müssen. Im Gegensatz dazu kann ein Föderationsknoten eine permanente, zuverlässige Netzwerkverbindung mit hoher Bandbreite nutzen, um mit den Komponenten der Dienstebene zu kommunizieren.

Cache-Konsistenzsemantiken

In der Diplomarbeit werden Lösungen zu folgenden drei Cache-Konsistenzsemantiken vorgestellt:

- **Strong:** Ein Cache-Inhalt stimmt immer zu 100% mit den Ursprungsdaten überein.
- **Delta**: Ein Cache-Inhalt kann für die Dauer von Δ-Zeiteinheiten veraltet sein. Folglich reflektiert eine Ergebnismenge im Cache einen Datenbankzustand, der vor Δ-Zeiteinheiten gültig war.
- **Weak**: Die Konsistenz eines Cache-Inhalts ist unbekannt. Eine zwischengespeicherte Ergebnismenge reflektiert in diesem Fall einen gültigen Datenbankzustand aus der Vergangenheit.

Erhaltung der Cache-Konsistenz eines Föderationsknotens

Die drei Cache-Konsistenzsemantiken lassen sich auf Seiten eines Föderationsknotens mit Hilfe von Strategien realisieren, die von Lösungsansätzen abgeleitet worden sind, die beim Caching im Web [4] sowie beim Caching in Client-Server Datenbanksystemen [2], [3] eingesetzt werden. Zur Umsetzung der Konsistenzstufe Weak wird ein *Validierungsansatz* verwendet, bei dem ein Föderationsknoten Datenänderungen von einem Spatial Model Server zu bestimmten Zeitpunkten abfragt. Bei der Konsistenzstufe Delta wird ein *hybrider Ansatz* eingesetzt, bei dem Datenänderungen sowohl von einem Föderationsknoten heruntergeladen, als auch von einem Spatial Model Server automatisch an einen Föderationsknoten propagiert werden. Eine effiziente Umsetzung der Konsistenzstufe Strong ist mit einem *Server-Invalidierungsansatz* möglich, bei dem Föderationsknoten über Datenänderungen, die ihren Cache-Inhalt betreffen, automatisch von einem Spatial Model Server informiert werden.

Erhaltung der Cache-Konsistenz mobiler Clients

Bei der Erhaltung der Cache-Konsistenz mobiler Clients hat sich ein Lösungsansatz bewährt, der von Strategien abgeleitet worden ist, die beim Caching im Web in drahtlosen Systemumgebungen eingesetzt werden [1]. Hierbei propagiert die Föderations-Middleware zu periodisch festgelegten Zeitpunkten Datenänderungen per *Broadcast* an alle verfügbaren mobilen Clients. Dabei wird angenommen, dass ein Föderationsknoten je einer Basisstation des drahtlosen Zugangsnetzes zugeordnet ist. Auf diese Weise können die Cache-Inhalte beliebig vieler mobiler Clients, die sich innerhalb der Reichweite der Basisstation befinden, gleichzeitig aktualisiert werden, was diesen Ansatz skalierbar macht. Damit keine irrelevanten Datenänderungen propagiert werden, verwaltet die Föderations-Middleware semantische Beschreibungen, welche die Cache-Inhalte mobiler Clients repräsentieren. Die Föderations-Middleware sichert einem mobilen Client zu, dass er über alle Datenänderungen informieren wird, die seinen Cache-Inhalt betreffen. Der Zeitpunkt der Propagation dieser Informationen hängt allerdings von der zu Grunde gelegten Cache-Konsistenzsemantik ab.

Literatur

[1] G. Cao. A Scalable Low-Latency Cache Invalidation Strategy for Mobile Environments. *IEEE Transactions on Knowledge and Data Engineering*, 15(5): 1251-1265, 2003.

[2] T. Härder and A. Bühmann. Datenbank-Caching – Eine systematische Analyse möglicher Verfahren. *Informatik – Forschung und Entwicklung*, 2004.

[3] A. Keller and J. Basu. A Predicate-Based Caching Scheme For Client-Server Database Architectures. *The VLDB Journal*, 5: 35-47, 1996.

[4] M. Rabinovich and O. Spatscheck. *Web Caching and Replication.* Addison-Wesley, 2002.

[5] http://www.nexus.uni-stuttgart.de