Grundlagen von Replikationstechniken für mobile Datenbanken*

Hagen Höpfner, *hoepfner@iti.cs.uni-magdeburg.de*

Institut für Technische und Betriebliche Informationssysteme
Otto-von-Guericke-Universität Magdeburg
Postfach 4120, 39016 Magdeburg

Zusammenfassung

In den letzten Jahren war ein stetiger Anstieg der Nutzerakzeptanz mobiler Telekommunikationsmedien zu beobachten. Wie die erzielten Preise der Versteigerung der UMTS-Lizenzen im Herbst 2000 deutlich gezeigt haben, sind auch die Telekommunikationsanbieter stark an mobilen Technologien interessiert.
Datenbanksysteme bieten zahlreiche Techniken zur Verwaltung und Manipulation von Daten (z.B. Transaktionen). Der weitere Ausbau des mobilen Telekommunikationsnetzes und die Entwicklung immer leistungsfähigerer und benutzerfreundlicherer Hardware bedingen den vermehrten Einsatz solcher Technologien und die Zunahme der zu verarbeitenden Nutzdatenmenge. Um eine redundante oder inkonsistente Datenhaltung zu verhindern, ist auch im mobilen Umfeld Replikation und der Einsatz von Datenbanken unumgänglich. Problematisch ist, daß mobile Knoten nicht permanente mit dem Netz verbunden sind. Die Gründe dafür sind verschieden. Zum einen können Empfangs- bzw. Sendeprobleme zu einer temporalen Nichterreichbarkeit führen, zum anderen kann der Nutzer das mobile Gerät abschalten.
In dieser Arbeit wird untersucht, inwieweit klassische verteilte Replikationsansätze im mobilen Umfeld eingesetzt werden können. Desweiteren werden spezielle Replikationsverfahren für mobile Datenbanken vorgestellt.

1 Einleitung

Maximale Mobilität ist in den meisten Berufen unumgänglich. Auch im privaten Umfeld kann ein stetiger Anstieg der Nutzerakzeptanz mobiler Telekommunikationsmedien beobachtet werden. Neben den Anwendern haben auch die Telekommunikationsanbieter starkes Interesse an mobilen Technologien. Dies wird insbesondere an den erzielten Preisen bei der Versteigerung der UMTS-Lizenzen im Herbst 2000 und dem Anteil mobiler Informationstechniken am Ausstellungs- und Vortragsprogramm der CeBIT 2001 deutlich.
Diese Aspekte und die fortlaufende Zunahme an Nutzdaten bedingen den vermehrten Einsatz mobiler Datenbank. Da solche Systeme besonderen Einschränkungen unterliegen (Speicherplatz, Übertragungsraten etc.), sind spezielle Techniken zum Umgang mit Daten in mobilen Umgebungen notwendig. In diesem Papier werden die Grundlagen für eine solche Technik, die Replikation, diskutiert.

2 Begriffe

In diesem Abschnitt werden grundlegende Begriffe eingeführt und die Betrachtung von Replikationstechniken für verteilte Datenbanken im mobilen Umfeld motiviert.
Nach [7] wird bei mobilen Datenbanken zwischen *mobilen Knoten* und *statischen Knoten* unterschieden. Desweiteren werden statische Knoten, welche die Einwahl von mobilen Knoten in das statische Netzwerk ermöglichen, als *Basisknoten* bezeichnet. Mobile Knoten werden aufgrund ihrer geographischen Lage in Bezug zu den Basisknoten in *Zellen* zusammengefaßt. Ein *Cluster* setzt sich aus allen Knoten zusammen, die sich in einer gemeinsamen geographischen Gebiet

*Diese Arbeit wird von der DFG gefördert (SA 782/3-1).

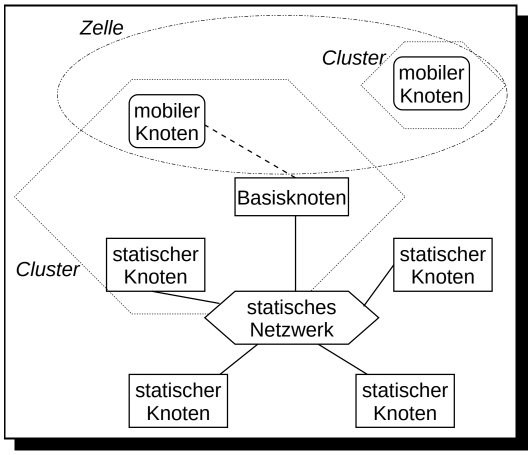

Abbildung 1: Nomenklatur der Knoten in mobilen Datenbanksystemen

befinden. Abbildung 1 illustriert diese Nomenklatur von Knoten in einem mobilen Datenbanksystem.
Die Kommunikation zwischen einem mobilen Knoten und einer Basisstation erfolgt über eine kabellose Funkverbindung. Daraus resultiert die Tatsache, daß nicht immer alle Knoten des Netzes erreichbar sind. Ist ein mobiler Knoten offline, wird er auch *disconnected node* genannt. Dieser Zustand kann durch verschiedene Einflüsse entstehen. Zum einen können Empfangs- bzw. Sendeprobleme dazu führen, daß der mobile Knoten nicht erreichbar ist. Zum anderen kann der Nutzer des mobilen Knotens diesen einfach ausschalten. Mobile Datenbanksysteme müssen die dadurch entstehenden Probleme lösen.

Mobile Datenbanksysteme sind verteilte Datenbanksysteme, in welchen mobile Knoten eingebunden sind [4]. Aus diesem Grund ist es naheliegend, zu untersuchen, inwieweit Replikationstechniken verteilter, nicht-mobiler Datenbanken bei mobilen Datenbanken eingesetzt werden können.
Replikation in verteilten Datenbanken umfaßt nach [2] die vier Bereiche Kopie-Update-Strategie (Datensynchronisation), Fehlerbehandlung, Synchronisation konkurrierender Zugriffe und Behandlung von Lesetransaktionen. In den folgenden Betrachtungen wird auf die ersten beiden Aufgaben der Replikation fokussiert. Kopie-Update-Strategien werden eingesetzt, um im fehlerfreien Betrieb Änderungen auf einem Knoten auch auf allen anderen betroffenen Replikaten durchzuführen. Der Punkt Fehlerbehandlung beinhaltet das Vorgehen im Fehlerfall um die Datenintegrität zu wahren bzw. wieder herzustellen. Auf entsprechende Ansätze wird in Abschnitt 3.1 bzw. Abschnitt 3.2 eingegangen.

3 Replikation in verteilten Datenbanken

In verteilten Systemen wird Replikation eingesetzt, um eine bessere Fehlertoleranz und einen höheren Datendurchsatz zu ermöglichen [1]. In verteilten Datenbanken gilt diese Zielsetzung auch. Wie in Abschnitt 1 erwähnt, wird bei den Betrachtungen zur Replikation in verteilten, nicht-mobilen Datenbanken auf die Datensynchronisation und die Behandlung von Konflikten fokussiert. In diesem Abschnitt werden kurz Techniken für diese Aufgabenbereiche vorgestellt.

3.1 Datensynchronisation

Laut [2] wird unterschieden, ob ein Verfahren alle Replikate parallel oder sequentiell über eine bestimmte Kopie aktualisiert. Techniken, welche alle Kopien auf einmal aktualisieren werden *read-*

one-write-all-Verfahren (ROWA) genannt. Die zweite Variante gliedert sich in Primär-Kopie- und Quorum-Verfahren. Beim Primär-Kopie-Ansatz werden Aktualisierungen eines Datenobjektes zuerst auf dessen Master-Kopie durchgeführt und durch diesen propagiert. Bei quorumbasierter Replikation wird durch mehrere Replikate entschieden, ob eine Änderung durchgeführt werden darf, oder nicht.

3.2 Behandlung von Netzpartitionierungen

Bei einer Netzpartitionierung zerfällt ein Netz in zwei Teilnetze, zwischen denen keine Kommunikation mehr möglich ist. Laut [2] wird hierbei zwischen progressiven und konservativen Verfahren unterschieden. Bei progressive Ansätze liegt die Priorität bei der Verfügbarkeit der Daten. Das heißt, daß auf den getrennten Teilnetzen weiter gearbeitet werden darf. Aufgetretene Inkonsistenzen werden hierbei in Kauf genommen. Wird die Verbindung zwischen den Teilnetzen wieder hergestellt, müssen die entsprechenden Datenbestände synchronisiert werden. Bei konservativen Techniken steht hingegen die Datenkonsistenz im Vordergrund. Daher darf nur auf einer Partition weitergearbeitet werden. Die Wahl dieser Hauptpartition erfolgt entweder aufgrund der Primär-Kopie eines Datenobjektes oder über Quorum-Verfahren.

4 Replikation in mobilen Datenbanken

Die in Abschnitt 3 vorgestellten Replikationstechniken können im mobilen Umfeld nur bedingt eingesetzt werden. Die Notwendigkeit für spezielle Verfahren ergibt sich aus den besonderen Anforderungen an mobilen Datenbanken und deren Eigenschaften. Mobile Knoten sind in der Regel offline und werden nur zur Datensynchronisation mit dem Netz verbunden. Das heißt, daß meistens eine Netzpartitionierung vorliegt. Da somit nur selten alle Knoten erreichbar sind, sind ROWA-Ansätze in mobilen Datenbanken nicht anwendbar. Ein mobiler Knoten wählt sich zur Synchronisation bei *einer* Basisstation ein. Da somit immer ein Quorum von 100% gegeben ist, ist der Einsatz von Quorum-Verfahren nicht notwendig.
Im Gegensatz dazu, bilden Primär-Kopie-Ansätze und progressive Verfahren die Grundlagen für Replikationsverfahren für mobile Datenbanken. Die folgenden Betrachtungen beschreiben einige solcher erweiterten Verfahren.

4.1 Provisorische Transaktionen

Eine Möglichkeit lokales Offline-Arbeiten mit mobilen Knoten zu ermöglichen, ohne die Integrität der gesamten Datenbank zu gefährden, ist die Verwendung von provisorischen Transaktionen. Diesen Ansatz verfolgt beispielsweise die Zwei-Schichten-Replikation [3], welche ein erweitertes Primär-Kopie-Verfahren ist. Befindet sich ein mobiler Knoten offline, werden alle Transaktionen provisorisch durchgeführt. Die dabei enstehenden Daten gelten auch als provisorisch. Provisorische Transaktionen unterliegen sogenannten Bereichsregeln, welche den Gültigkeitsbereich dieser Transaktionen definieren. Beispielsweise kann mit einer Bereichsregel sichergestellt werden, daß eine Abbuchung nur dann global durchgeführt werden darf, wenn dadurch der Kontostand nicht negativ wird. Bei der Wiedereinwahl des mobilen Knotens werden die provisorischen Transak-

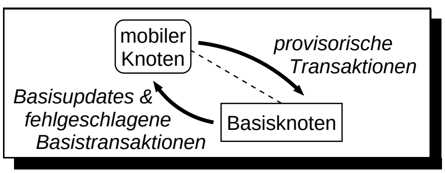

Abbildung 2: Zwei-Schichten-Replikation: Informationsfluß

tionen auf die Basisstation übertragen. Dort wird versucht, diese Transaktionen global unter Beachtung der Bereichsregeln auszuführen. Wie in Abbildung 2 dargestellt ist, werden die entsprechenden Ergebnisse von der Basistation an den mobilen Knoten zurück propagiert. Schlägt eine provisorische Transaktion bei der globalen Ausführung fehlt, erhält der mobile Knoten eine Fehlermeldung. Anderenfalls wird diese Transaktion in eine Basistransaktion überführt.
Die Propagation der Transaktionen an die anderen Knoten des statischen Netzes erfolgt nach einem Standard-Primär-Kopie-Verfahren.

Ein weiteres Beispiel für die Verwendung von provisorischen Transaktionen ist das Bayou-System [9]. Bei der Synchronisation eines mobilen Knotens mit einer Basisstation werden in diesem System weitere Techniken zur Konflikterkennung bzw. -behebung eingesetzt. Dazu werden die Schreiboperationen um *per-write dependency checks* und *per-write merge-procedures* erweitert [8]. Wird bei der Synchronisation durch den Abhängigkeitstest ein Konflikt festgestellt, wird die Mischfunktion aufgerufen und versucht den Konflikt zu lösen. Ist dies nicht möglich, erfolgt eine Fehlermeldung.

4.2 Snapshots

Neben provisorischen Transaktionen (vgl. Abschnitt 4.1) kann die Haltung von verschiedenen Versionen der Daten zur Konfliktbehebung bei der Synchronisation eingesetzt werden. Das folgende Beispiel [6] illustriert die Funktionsweise von Snapshot-Verfahren.
Gegeben sei der folgender Ablauf auf der Basistation:

1. Schreiben der beiden Attribute $x = 1$ und $y = 1$ und Abschließen der Transaktion
2. Lesen der Werte der beiden Attribute
3. Schreiben von $x = 2$ und Abschließen der Transaktion

Der mobile Knoten liest die Daten nach der Bestätigung der ersten Transaktion und geht offline. Zu diesem Zeitpunkt haben x und y den Wert 1. Danach schreibt der mobile Knoten lokal das Attribut $y = 3$ und bestätigt die entsprechende Transaktion.
Bei der nächsten Einwahl des mobilen Knotens muß diese Transaktion auf dem Basisknoten global ausgeführt werden. Offensichtlich besteht hierbei ein Konflikt (auf Basisknoten: $x = 2$ und $y = 1$; Transaktion auf mobilem Knoten hat $x = 1$ und $y = 1$ gelesen). Bei der Verwendung von Snapshots, sind beide Versionen der Datenobjekte (Version 1 : $\{x_0 = 1, y_0 = 1\}$, Version 2 : $\{x_1 = 2, y_0 = 1\}$) gespeichert. Diese Informationen können genutzt werden, um die Transaktion so umzuschreiben, daß die Konflikte behoben sind. Das Ergebnis sieht folgendermaßen aus:

1. Schreiben der beiden Attribute $x = 1$ und $y = 1$ und Abschließen der Transaktion
2. Lesen der Werte beiden Attribute $x = 1$ und $y = 1$ und der Werte dieser Attribute auf dem Client $(x = 1, y = 1)$
3. Übertragen des neuen y-Wertes vom Client auf den Server $(y = 3)$ und Abschließen der Transaktion
4. Schreiben von $x = 2$ und $y = 3$ und Abschließen der Transaktion

Dieser Ansatz funktioniert nur, wenn sich der y-Wert auf dem Basisknoten nicht geändert hat. Ein Verfahren, welches diesen Ansatz verfolgt und um zusätzliche Konflikbehebungs-Funktionalität erweitert, ist der Multiversion-Reconciliation-Algorithmus [5]. Hierbei werden Konfliktbehebungsfunktionen eingeführt, welche abhängig von einer Kostenfunktion (Kostenminimum) die Synchronisation zwischen Snapshots durchführen, wenn dadurch keine neuen Konflikte mit anderen Snapshots auftreten.

5 Fazit und Ausblick

Replikationstechniken für verteilte, nicht-mobile Datenbanken können nur bedingt im mobilen Umfeld eingesetzt werden. Die Verwendung von ROWA-Techniken ist sowohl online, als auch im Offline-Betrieb nicht möglich. Ist ein mobiler Knoten am Netz angemeldet, können normale Primär-Kopie-Ansätze zur Änderungspropagation genutzt werden. Progressive Techniken zur Behandlung von Netzpartitionierungen bilden die Grundlage für Verfahren zur Synchronisation eines mobilen Knotens mit einem Basisknoten nach der Wiedereinwahl des mobilen Knotens.

Im Rahmen des DFG-Projektes „Adaptive Replikation von Daten in heterogenen mobilen Kommunikationsnetzen“ wird untersucht, inwieweit die in diesem Papier vorgestellten Replikationstechniken um Aspekte der Adaptivität erweitert werden können. Außerdem wird der Einsatz von Techniken zur Konfliktbehebung und -behandlung aus dem Bereich der Datenförderierung im Kontext der Synchronisation von Replikation betrachtet. Ein weiterer Gesichtspunkt in diesem Projekt ist die Replikation zwischen mobilen Knoten. Es wird untersucht, in welchen Fällen die Verwendung von Basisstationen notwendig ist und unter welchen Bedingungen direkt zwischen den mobilen Knoten kommuniziert werden kann.

Literatur

[1] T. Beuter and P. Dadam. Prinzipien der Replikationskontrolle in verteilten Datenbanksystemen. *GI Informatik Forschung und Entwicklung*, 11(4):203–212, November 1996.

[2] P. Dadam. *Verteilte Datenbanken und Client/Server-Systeme — Grundlagen, Konzepte, Realisierungsstrukturen*. Springer-Verlag, Berlin, 1996.

[3] J. Gray, P. Helland, P. O'Neil, and D. Shasha. The dangers of replication and a solution. *SIGMOD Record*, 25(2):173–182, 1996.

[4] H. Höpfner. Replikationstechniken für mobile Datenbanken. Diplomarbeit, Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik, Institut für Technische und Betriebliche Informationssysteme, March 2001.

[5] Shirish H. Phatak and B. R. Badrinath. Multiversion reconciliation for mobile databases. In *Proceedings of the 15th International Conference on Data Engineering, 23-26 March 1999, Sydney, Austrialia*, pages 582–589. IEEE Computer Society, March 1999.

[6] Shirish H. Phatak and B.R. Badrinath. Transaction-centric reconciliation in disconnected databases. *ACM Monet Journal*, 53, 1999.

[7] E. Pitoura and B. Bhargava. Maintaining consistency of data in mobile distributed environments. In *Proceedings of the 15th International Conference on Distributed Computing Systems*, pages 404–413, May 1995.

[8] D. B. Terry, M. M. Theimer, K. Petersen, A. J. Demers, M. J. Spreitzer, and C. Hauser. Managing update conflicts in bayou, a weakly connected replicated storage system. In *Proceedings 15th Symposium on Operating Systems Principles (SOSP-15)*, pages 172–183, Cooper Mountain, Colorado, December 1995. Association for Computing Machinery, Inc.

[9] M. M. Theimer, A. J. Demers, K. Petersen, M. J. Spreitzer, D. B. Terry, and B. B. Welch. Dealing with tentative data values in disconnected work groups. In *Proceedings of the Workshop on Mobile Computing Systems and Applications*, pages 192–195, Santa Cruz, California, December 1994. Institute of Electrical and Electronics Engineers.