VROBS – Datenbankzugangskontrolle mittels einer verteilten Warteschlange

Lars Bindzus, Klemens Böhm
GMD - Forschungszentrum Informationstechnik GmbH
Institut für Integrierte Publikations- und Informationssysteme
(GMD-IPSI)
E-mail: {bindzus,kboehm}@darmstadt.gmd.de

Zusammenfassung

Wenn Datenbankzugriff über das Internet ermöglicht werden soll, ist es sinnvoll, eine bestimmte Anzahl von Datenbank-Klienten vorrätig zu halten, die auf Messages und Anfragen warten, die sie dann abarbeiten. Die Effizienz des Gesamtsystems hängt von der Strategie ab, nach der die Zuweisung von Messages und Anfragen an die Klienten erfolgt. Das Ziel unserer Arbeit ist es, einen Rahmen zu schaffen, in dem für einen vorgegebenen Kontext gute Strategien ermittelt werden können. Die Architektur wird dafür auf ein erweitertes Mehrkanalmodell abgebildet.

Kategorien: a, c

1 Einleitung

Ein großer Teil der derzeit im Internet verfügbar gemachten Informationen ist in Form von HTML-Dokumenten in Files auf WWW-Servern enthalten. Es ist jedoch sinnvoll, wenn über das Internet auch auf Datenbankinhalte zugegriffen werden kann. Zu den Datenbankfeatures, die in diesem Zusammenhang vorteilhaft sind, gehören - insbesondere bei objektorientierter Datenbanktechnologie - ausdrucksstarke, deklarative Zugriffsmechanismen sowie die Möglichkeit, die Anwendungssemantik zu modellieren und die Konsistenz der Daten sicherzustellen.

Der nebenläufige Datenbankzugriff über das Internet kann folgendermaßen realisiert werden:

- Für die Abarbeitung einer Anfrage bzw. Message wird ein Bearbeitungsprozeß vom Datenbank-Server geforkt. Problematisch ist jedoch die Tatsache, daß das Forken die Abarbeitung der Operation verzögert. Dieses Phänomen wird im folgenden als *DBMS-Geschwindigkeitsproblem* bezeichnet.

- Das DBMS-Geschwindigkeitsproblem läßt sich umgehen, indem eine bestimmte Anzahl von Datenbank-Klienten bereits gestartet ist, die auf abzuarbeitende Anfragen warten, und ggf. ihnen zugegangene Anfragen bis zur Abarbeitung in einer Warteschlange verwalten. Die eingehenden Queries und Messages werden nach einer bestimmten Strategie auf die Klienten verteilt.

Der zweite Ansatz wird im folgenden betrachtet. Das Problem, mit dem wir uns hier beschäftigen, ist die Frage, nach welcher Strategie die eingehenden Anfragen und Messages verteilt werden, um Wartezeiten möglichst gering zu halten. Eine allgemeingültige Antwort existiert offensichtlich aus den folgenden Gründen nicht:

- Die Entscheidung, welchem Klient eine Message zugewiesen werden sollte, hängt von der zu erwartenden Dauer der Operation ab, d.h. von ihrem Typ und ihren Parametern. Um die richtige Entscheidung zu treffen, ist im Prinzip Wissen über die Anwendungssemantik erforderlich.

- Die Vorgabe "möglichst geringe Wartezeiten" ist nicht eindeutig. Es kann einerseits angestrebt werden, die Gesamtwartezeit aller Teilnehmer zu minimieren. Es gibt aber auch andere mögliche Ziele, z.B. die maximale Wartezeit, die auftreten darf, möglichst gering zu halten.

Unser Vorgehen ist das folgende: Die Gesamtarchitektur, die für das OODBMS VODAK [VML95] entwickelt wurde, enthält eine Komponente, den VROBS ('VODAK Remote API Observer'), der die Zuweisung der Messages an die Klienten vornimmt. Diese Komponente ist so konzipiert, daß die Zuweisungsfunktion leicht ausgetauscht werden kann. Der Ansatz, den wir verfolgen, ist der, unterschiedliche Zuweisungsfunktionen zu realisieren und zu testen. Insbesondere interessiert uns die Fragestellung, ob ausgefeilte Zuweisungsfunktionen deutliche Vorteile bringen.

Der Implementationsaufwand von entsprechenden Zuweisungsfunktionen ist zum Teil erheblich. Deshalb ist eine Implementation dieser Funktionen ohne vorherige Untersuchung nicht sinnvoll. Die Schwierigkeit besteht nun darin, Aussagen über die zu erwartende Performanz komplexer Zuweisungsfunktionen zu treffen und eine Auswahl der zu implementierenden Funktionen festzulegen.

Wir diskutieren drei unterschiedliche Zuweisungsfunktionen: Nach einer Beschreibung der Funktionsweise zeigen wir die Einsetzbarkeit jeder Funktion auf. Schließlich treffen wir auf der Grundlage warteschlangentheoretischer Ergebnisse Aussagen über die zu erwartende Performanz und charakteristische Eigenschaften, soweit dies analytisch möglich ist.

In diesem Artikel wird zum einen unsere Architektur beschrieben, und zwar in Abschnitt 2. Wir skizzieren ferner das entsprechende Warteschlangenmodell, das es uns erlaubt, die optimale Konfiguration zu ermitteln, wenn bestimmte Kenngrößen als bekannt angenommen werden. In Abschnitt 3 erläutern wir die unterschiedlichen Zuweisungsfunktionen, die grundsätzlich möglich sind, und die bei den Experimenten berücksichtigt werden sollen. An gleicher Stelle stellen wir unsere Schlußfolgerungen über die Qualität dieser Funktionen vor.

Der gegenwärtige Status unserer Arbeit ist die Erstellung eines Prototyps, so daß wir derzeit noch keine empirischen Aussagen im Rahmen einer umfassenden Statistik treffen können.

2 Architektur

Im folgenden wird die Architektur ausführlicher erläutert, in die der VROBS eingebettet ist. Dabei wird nach Darstellung der Einzelkomponenten und spezifischer Phänomene das Gesamtsystem auf das Modell einer verteilten Warteschlange übertragen. Schließlich erfolgt die Integration einer zusätzlichen DBMS-Kontrollkomponente in den VROBS.

Die Einzelkomponenten des Systems und ihr Zusammenspiel. Das Gesamtsystem besteht im wesentlichen aus den fünf Komponenten HTTP-Server, HTTP-Applikationen, VROBS, Datenbank-Klienten und DBMS inklusive dessen geforkter Prozesse. Abbildung 1 gibt eine schematische Übersicht für ein Szenario mit VROBS und n Datenbank-Klienten. Die Abarbeitung einer Anfrage besteht aus den folgenden Schritten: Ein WWW-Klient stellt an eine Datenbank eine Anfrage über den HTTP-Server, der dafür eine neue HTTP-Applikation startet. Diese Applikation verbindet sich nun nicht direkt mit dem DBMS als eigenständiger Klient. Stattdessen meldet sie sich beim VROBS an (1). Die Kommunikation mit der Datenbank findet ausschließlich über die durch den VROBS verwalteten Datenbank-Klienten statt. Der Observer teilt der Applikation einen Datenbank-Klienten zu (2). Die Applikation wickelt die weitere Datenbank-Kommunikation über ihn ab (3). Nach Beendigung meldet sich die Applikation beim VROBS ab. Der VROBS registriert dies, um künftige Anfragen sinnvoll auf die Klienten verteilen zu können (4).

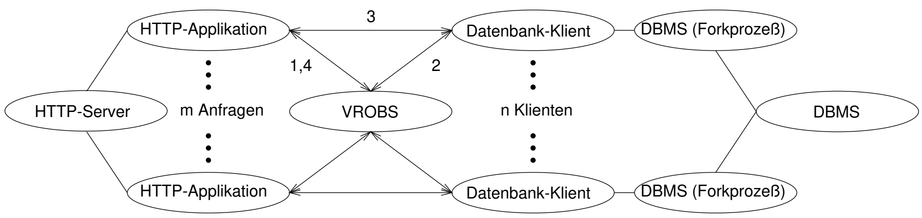

Abbildung 1: Gesamtsystem mit n Datenbank-Klienten und Observer

Wie in der Einleitung angedeutet, wird bei einer Klient-Verbindung der Bearbeitungsprozeß des Datenbank-Servers geforkt. Bei Anfragen an das System mit einer solchen Klient-Verbindung mit kurzer Bearbeitungsdauer wird die Gesamtantwortzeit wesentlich durch den Forkpunkt für den Bearbeitungsprozeß im DBMS bestimmt. Wird früh geforkt, müssen u.U. innerhalb des geforkten Prozesses diverse Systemtabellen initialisiert werden. Die Verzögerung, die sich durch die resultierende Initialisierungszeit ergibt, kann insbesondere bei zeitkritischen Anwendungen Probleme mit sich bringen. Man möchte deshalb den Forkpunkt hinter sämtliche Initialisierungsroutinen legen. Nicht nur bei Datenbankzugriffen über das WWW, bei denen sich ein hoher Parallelitätsgrad ergeben kann, ist man an einer Lösung des Forkpunkt-Problems interessiert. Die VROBS-Architektur löst das Problem dadurch, daß sämtliche Verbindungen zum DBMS schon im voraus hergestellt worden sind. D.h. sämtliche Datenbank-Klienten sind aktiv und warten ihrerseits auf Anfragen von HTTP-Applikationen.

Die Architektur als verteiltes Warteschlangenmodell. Die Tatsache, daß wir uns für die Realisierung der VROBS-Architektur entschieden haben, resultiert aus unserer Absicht, eine möglichst effiziente Verteilung eingehender Anfragen auf die einzelnen Datenbank-Klienten zu erreichen. Dazu wird die Architektur als eine verteilte Warteschlange aufgefaßt, d.h. das Gesamtsystem bildet ein erweitertes Mehrkanalmodell, bzw. handelt es sich um eine Mischung mit einem seriellen Mehrkanalmodell. In einem Mehrkanalmodell gibt es eine Warteschlange und n Bedienstationen. Unsere Erweiterung sieht so aus, daß die Bedienstationen selbst Warteschlangen haben. Dies ist in Abbildung 2 zu sehen. Die linken Hälften der Rechtecke symbolisieren jeweils eine Warteschlange, die rechten die Bedienstationen.

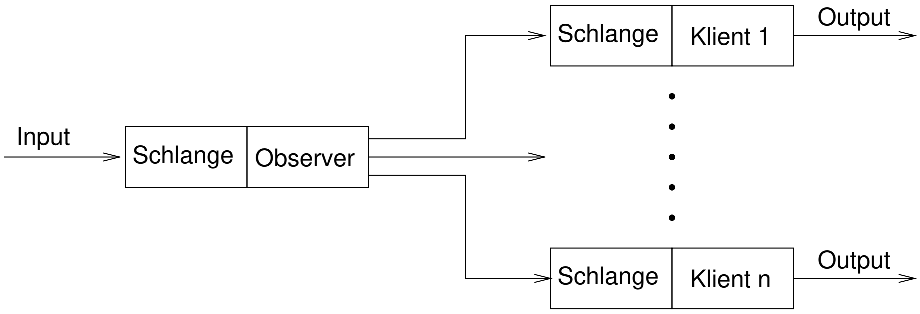

Abbildung 2: Gesamtsystem als verteilte Warteschlange

Der globale Warteraum wird durch eine vorgeschaltete Warteschlange mit dem VROBS als Bedienstation erster Stufe und dann den einzelnen Warteschlangen mit den Datenbank-Klienten als Bedienstationen gebildet. Eingehende Anfragen gelangen zuerst in die globale Warteschlange. Von dort nimmt der Observer die Verteilung auf die einzelnen Bedienstationen vor, d.h. die Adressenzuweisung. Die Anfragen gelangen in die verteilten Warteschlangen und werden nach einer Wartezeit vom zugehörigen Datenbank-Klient bearbeitet.

Die Betrachtung der Architektur als verteiltes Warteschlangenmodell erlaubt eine quantitative Analyse der Performanz des Gesamtsystems. Nach Feststellung der Kenngrößen, d.h. insbesondere Verteilungsarten und Parameter für Ankunftszeiten, Bedienzeiten und Anzahl der Bedienstationen kann analytisch oder simulativ eine Abschätzung zwischen Rechnerleistung und Anzahl paralleler Datenbank-Klienten, VROBS und DBMS durchgeführt werden. Entsprechende Formeln, die unter Zuhilfenahme vereinfachender Annahmen direkt angewendet werden können, finden sich in [HL88, DD91].

Charakteristisch für die Architektur sind eine konstante Bedienzeit des VROBS, d.h. eine konstante Zuteilungszeit von Adressen, variable Bedienzeiten der Datenbank-Klienten, die abhängig sind vom Bearbeitungsaufwand des DBMS, kurze Wartezeiten in der globalen Warteschlange, die einem Vielfachen der konstanten Bedienzeit des VROBS entsprechen, und je nach Zuteilungsstrategie stark differierende Wartezeiten in den einzelnen Warteschlangen.

Die Gesamtarchitektur hat neben einem verlängerten Verwaltungsaufwand die Eigenschaft, daß der VROBS nach Zuteilung einer Anfrage in die entsprechende Warteschlange die Kontrolle über diese Anfrage verliert, d.h. er kann sie im Nachhinein nicht mehr umsortieren. Andererseits bietet dieses System einer verteilten Warteschlange folgende Vorteile:

- Einmal zugeteilt, wird eine Anfrage auf jeden Fall beantwortet, ohne daß sie noch einmal zurückgezogen wird, falls eine spätere Anfrage höherer Priorität eintrifft.
- Die gemeinsame Warteschlange des VROBS benötigt nur einen kleinen Puffer, da eingehende Anfragen umgehend auf die anderen Warteschlangen verteilt werden.
- Da Betriebssystem-Mechanismen für die Realisierung der Warteschlange verwendet werden, ist die Implementierung einfach. Darüber hinaus brauchen eventuell auftretende Timeout-Probleme eingetroffener Anfragen nicht auf der Ebene des VROBS gelöst zu werden. Entsprechende Mechanismen sind ebenfalls Teil des Betriebssystems.

Der VROBS als Kontrollkomponente. Neben der Zuteilung der Anfragen übernimmt der VROBS die Kontrolle der Datenbank-Klienten und des DBMS. Dabei sind drei verschiedene Tasks vorgesehen: Start der Datenbank-Klienten, periodische Integritätskontrolle des DBMS und der Datenbank-Klienten und, falls erforderlich, Neustart von DBMS und Datenbank-Klienten. 'Integrität' bzw. 'Konsistenz' bedeutet hier, daß der Server bzw. die Klienten noch richtig arbeiten, und nicht abgestürzt sind oder sich in einem undefinierten Zustand befinden.

Diese von der Zuteilungsaufgabe unabhängigen Tasks könnten prinzipiell auch von einer separaten Komponente erledigt werden. Der Vorteil davon, daß die Kontrolle in den Observer integriert ist, liegt aber darin, daß der Observer einerseits ein sofortiges Feedback über nichtkonsistente Datenbank-Klienten erhält und andererseits eine u.U. aufwendige Kommunikation zwischen zwei voneinander unabhängigen Komponenten entfällt.

3 Message-Zuordnungsfunktionen und ihre Charakterisierung

Ein wichtiger Faktor für eine effiziente Bearbeitung der ankommenden Anfragen ist die Zuteilungsstrategie auf die Datenbank-Klienten. Neben einer möglichst kurzen Wartezeit für jede einzelne Anfrage kann das Ziel auch sein, die Gesamtwartezeit aller Anfragen zu minimieren. Die Zuteilungsstrategien, die im folgenden beschrieben werden, bieten sich grundsätzlich an.

1. **Streng zyklische Zuteilung.** Der VROBS geht die Datenbank-Klienten der Reihe nach durch. Wenn er auf einen Klienten mit leerer Warteschlange trifft, ordnet er diesem die nächste Anfrage zu. Wenn kein Klient eine leere Warteschlange hat, werden die eingehenden Anfragen streng zyklisch auf die Klienten verteilt. - Dieses Vorgehen erscheint weniger effektiv als die folgenden Varianten, erfordert jedoch andererseits keinerlei "Intelligenz" des VROBS und ist leicht implementierbar.

2. **Zuteilung zur geschätzten kürzesten Warteschlange.** Der Observer hat Kenntnis der Applikationssemantik und kann somit Aussagen über den zu erwartenden Bearbeitungsaufwand einzelner Anfragen an das DBMS treffen. Er wählt den Datenbank-Klienten mit der gemäß seiner Schätzung kürzesten Warteschlange aus. - Diese Variante ist im Gedankenmodell die effektivste der Alternativenn, erfordert jedoch Wissen über die Anwendung.

3. **Zuteilung unterschiedlicher Messagetypen zu unterschiedlichen Warteschlangen.** Der VROBS kann aufgrund der Herkunft der Anfragen einzelne Anfragetypen voneinander unterscheiden. Darüber hinaus können verschiedene Anfragetypen entweder (a) getrennten Warteschlangen zugeordnet werden, oder (b) es werden ihnen unterschiedliche Prioritäten zugeordnet. Im Fall (a) werden für bestimmte Anfragetypen eine feste Anzahl von Datenbank-Klienten reserviert. Die Idee ist eine Verallgemeinerung des Konzepts 'Schnellkasse im Supermarkt', bei der nur Kunden mit wenig Waren abgefertigt werden. Ausgehend von einer Implementierung von Strategie 1 läßt sich diese Variante mit relativ geringem Mehraufwand realisieren. - Der Fall (b) erfordert eine Verwaltung der globalen Warteschlange des VROBS. Hier wird also von dem in dem vorigen Abschnitt beschriebenen Modell abgewichen. Anfragen höherer Priorität werden in der Schlange an vorderer Stelle einsortiert. Voraussetzung für diese Alternative ist das Erkennen des Anfragers (hier: HTTP-Applikation mit spezieller URL). Das ist natürlich eine Einschränkung. - Die Beantwortungszeiten für verschiedene Anfragetypen sind leicht ermittelbar, ohne daß Wissen über die Anwendungssemantik, wie bei Alternative 2 gegeben sein muß.

Aussagen über die Qualität der drei Arten Zuordnungsfunktionen sind schon vor Beginn einer Simulation durch eine theoretische Analyse möglich. Eine hohe Qualität wird dann erreicht, wenn die vier Kenngrößen für Warteschlangensysteme, nämlich Kundenanzahl im System, Wartezeit im System, Kundenanzahl in der Schlange und Wartezeit in der Schlange, gering sind. Zusätzlich hat jede Zuordnungsmethode spezifische Eigenschaften wie die Varianz der Beantwortungszeiten, der Implementationsaufwand, die Möglichkeit einer theoretischen Analyse und das gleichzeitige Auftreten mehrerer Anfragearten. Das Kriterium 'Gleichzeitiges Auftreten mehrerer Anfragearten' steht für die folgende Unterscheidung: Entweder hat man es mit Anfragen zu tun, deren Ankünfte und Bedienzeiten einer einzigen Verteilung folgen, d.h. es gibt nur einen Anfragetyp. Im anderen Fall handelt es sich um Anfragen unterschiedlicher Art, deren Ankunfts- und Bedienzeiten unterschiedliche Verteilungen aufweisen, man hat also mehrere Anfragetypen.

In [Wolff89] wird gezeigt, daß die Kenngrößen von Strategie 1 mit denen für eine gemeinsame Warteschlange übereinstimmen. Die zyklische Zuteilung hat eine höhere Varianz der Antwortzeiten zur Folge als bei gemeinsamer Warteschlange. Sie ist jedoch leicht implementierbar. Eine theoretische Analyse ist möglich, aber nur, wenn man sich auf Anfragen eines Typs beschränkt.

Solange es nur einen Anfragetyp gibt, sind die Kenngrößen bei Zuordnungsstrategie 2 identisch mit denen einer gemeinsamen Warteschlange. Die Varianz der Antwortzeiten ist dabei niedriger als bei Zuordnungsstrategie 1. Der Vorteil, verglichen mit Strategie 1, ist der, daß differenziert werden kann zwischen Anfragen unterschiedlicher Typen. Eine theoretische Analyse ist weiterhin möglich für einzelne zu optimierende Größen [Wolff89]. Die Implementation ist jedoch entsprechend aufwendig.

Zuordnungsmethode 3 ist ein Beispiel für ein Prioritätensystem, für das eine theoretische Analyse nur eingeschränkt möglich ist [HL88, Wolff89]. Jedem Anfragetyp wird eine Priorität zugeordnet, so daß die Bearbeitungsreihenfolge in der Schlange für unterschiedliche Anfragen festgelegt ist. Bei einer gemeinsamen Warteschlange ist die Optimierung nach Gesamtwartezeit theoretisch analysierbar. Diese Optimierung benachteiligt jedoch die Anfragen, die eine lange Bearbeitungsdauer benötigen. Hat man unterschiedliche Warteschlangen für jeden Anfragetyp, so muß der Benutzer im voraus das durchschnittliche Aufkommen der einzelnen Anfragetypen ermitteln. Danach kann er die jeweilige Klientenanzahl festlegen, die den einzelnen Anfragearten zur Verfügung steht. Die Analyse für eine Art von Anfragen ist dann identisch mit der bei zyklischer Zuteilung. Methode 3 unter Verwendung getrennter Warteschlangen ist leicht realisierbar. Die Implementierung mit einer Warteschlange und Umsortierung eingehender Anfragen ist dagegen wesentlich aufwendiger.

4 Schlußbemerkungen

Um bei parallelen Datenbankzugriffen, die im WWW-Kontext vorkommen, das DBMS-Geschwindigkeitsproblem zu überwinden, wurde die VROBS-Architektur entwickelt. Dabei müssen die Datenbankzugriffe mit sehr guter Performanz zu einzelnen Datenbank-Klienten zugeteilt werden. Der Artikel zeigt, daß entsprechende Zuteilungsmechanismen auf der Grundlage warteschlangentheoretischer Analyse qualitativ eingeschätzt und ausgewählt werden können, und stellt drei dieser Mechanismen vor.

Dabei hat sich herausgestellt, daß bei gleichartigen Datenbankzugriffen die zyklische Zuordnung der Anfragen auf die einzelnen Klienten eine sinnvolle Lösung ist. Denn bei geringerem Implementationsaufwand wird eine ähnliche Performanz erreicht wie bei einer Zuordnung mit Kenntnis über die Applikationssemantik. Inwieweit bei heterogenen Anfragetypen die Zuordnungsmethode mit Wissen über die Applikationssemantik den Vorzug vor einer Methode mit Aufteilung unterschiedlicher Messagetypen auf unterschiedliche Warteschlangen erhält, hängt offensichtlich vom jeweiligen Optimierungsziel ab.

Neben einer umfassenden empirischen Analyse wird sich die zukünftige Arbeit an Fragen alternativer Optimierungsziele orientieren. Dafür läßt sich die zweite Message-Zuordnungsfunktion um KI–Mechanismen erweitern, indem der VROBS quasi als wissensbasierter Agent arbeitet. Er kann dann auf Grundlage seiner Wissensbasis über vergangene Anfrage-Pattern und Benutzervorgaben Schlüsse für zukünftige Zuteilungsentscheidungen ziehen.

Literatur

[DD91] Wolfgang Domschke and Andreas Drexl. *Einführung in Operations Research*. Springer-Verlag Berlin Heidelberg New York, 1991.

[HL88] Frederick Hillier and Gerald Lieberman. *Operations Research*. Oldenbourg München, Wien, 1988.

[MP90] Rudolf Mathar and Dietmar Pfeifer. *Stochastik für Informatiker*. Teubner Stuttgart, 1990.

[VML95] VODAK V 4.0 User Manual. Technical Report 910, GMD-IPSI, April 1995. St. Augustin.

[Wolff89] Ronald W. Wolff. *Stochastic Modeling and the Theory of Queues*. Prentice-Hall Englewood Cliffs, 1989.