Institut für Parallele und Verteilte Höchstleistungsrechner (IPVR)
Universität Stuttgart

MediaStore

Der MediaStore-Browser: Datenbankzugriffe auf multimediale Daten via World-Wide Web

Peter Kutschera

Zusammenfassung

Die Popularität des World-Wide Webs (WWW) basiert im wesentlichen auf der intuitiven Bedienbarkeit der graphischen Frontends, die den Zugriff auf weltweit verteilte Ressourcen ermöglichen. Aufgrund der allgemeinen Verfügbarkeit der WWW-Software auf sämtlichen Hardwareplattformen ist das World-Wide Web auch als Basis für andere Anwendungsbereiche wie z.B. Datenbankanwendungen interessant. Die Zielsetzung des MediaStore-Browsers stellt die Realisierung eines World-Wide Web basierten Datenbank-Browsers dar, der in der Lage sein sollte einen Zugriff auf textuelle (Relationen bzw. Objekte) und kontinuierliche Daten (Audio und Video) zu ermöglichen. Um die Anwendbarkeit des Browsers für ein breites Anwendungsspektrum zu gewährleisten, wurde die Funktionalität der Benutzerschnittstelle unabhängig vom zugrundeliegenden Datenmodell realisiert. Bei der Implementierung wurde insbesondere auf die Verwendung von Standards wie ODBC (Open Database Communication) und CGI (Common Gateway Interface) Wert gelegt.

Einführung

Die technologischen Entwicklungen und die sinkenden Hardwarepreise innerhalb der letzten Jahre führten nicht nur zu einer jährlichen Verdopplung der Prozessorleistung, sondern ebenfalls zu einer Erhöhung der Zugriffsgeschwindigkeit auf Tertiärspeicher und einer Entwicklung von Netzwerken mit Übertragungsbandbreiten im Gigabitbereich. Diese Leistungssteigerungen im Hardwarebereich führte zu der Situation, daß viele Anwendungen, die in der Vergangenheit nur mit teurer Spezialhardware realisiert werden konnten, einer wachsenden Zahl von Anwendern zugänglich gemacht werden konnten. Diese Entwicklung führte insbesondere im Bereich Multimedia in der Folgezeit zu einer Entwicklung der verschiedensten Applikationen wie Video-On-Demand, Audio- und Videoarchive, Lernsysteme, Systeme für kooperative Gruppenarbeit oder Konferenzsysteme, die von ihren jeweiligen Entwicklern als die “Killerapplikation“ des kommenden Multimediazeitalters bezeichnet wurde. Gleichzeitig wurde auch der Ruf nach einer Bereitstellung von schnelleren Netzwerken laut, um eine Übertragung von Audio- und Videodaten in Realzeit zu ermöglichen und so dem Anwender die Teilnahme an diesen multimedialen Diensten von zuhause aus zu ermöglichen. Daraufhin wurden von staatlicher Seite die verschiedensten Initiativen ins Leben gerufen, um dieses Defizit zu beseitigen. Die bekannteste Initiative dieser Art, stellt die in den USA angelaufende ``National Information Infrastructure“ (NII) dar, die sich zum Ziel gesetzt hat, alle Regierungsinstitutionen, Schulen, Universitäten, öffentliche Einrichtungen, sowie privatwirtschaftliche Firmen mit Hochgeschwindigkeitsnetzen (Information Highways) zu verbinden.

Einige Jahre zuvor, startete am European Center for Nuclear Research (CERN) in Genf ein Projekt, das fünf Jahre nach seinem Beginn die wohl populärste multimediale Anwendung hervorgebracht hat, und vielfach als die “Killerapplikation“ für Hochgeschwindigkeitsnetze angesehen wird. Das Ziel, des im Oktober 1990 begonnenen World-Wide Web Projekts (WWW), war die Realisierung eines verteilten Hypermediasystems. Die grundlegende Idee von Hypermediasystemen ist die Verwaltung von nicht-linearen Informationen, deren Grundelemente Texte, Graphiken, Audio, Video, usw. darstellen. Diese Grundelemente werden durch Querverweise, sogenannte Hyperlinks, miteinander verbunden, so daß sich ein Benutzer innerhalb eines Hypermediasystems navigierend entlang dieser Querverweise fortbewegen kann. Diese Entwicklung blieb aber weitestgehend unbeachtet, bis das National Center for Supercomputing Applications (NCSA) im Jahre 1993 den graphischen Browser Mosaic entwickelte. Dieser Browser ermöglichte den Zugriff auf das World-Wide Web mittels Unix-Workstations mit Internetzugang und setzte damit eine Entwicklung in Gang, die auch von den Entwicklern nicht vorausgesehen werden konnte. So wurden bereits im Juni 1994 bei einer Erhebung 1500 registrierte WWW-Server ermittelt. Im Jahr 1995, das auch als Jahr der Kommerzialisierung der WWW-Dienste bezeichnet werden kann, wurde die Popularität des World-Wide Webs vor allem durch Publikationen in den Massenmedien noch gesteigert. Diese Popularität führte mittlerweile zu einem Anstieg auf 7000 registrierte WWW-Server.

Aufgrund der allgemeinen Verfügbarkeit der WWW-Software (Server und Browser) auf sämtlichen Hardwareplattformen, der Möglichkeit zum weltweiten Zugriff auf Ressourcen und der einfachen Bedienbarkeit der Browser, ist es sinnvoll, das World-Wide Web auch als Basis für eine Reihe von Anwendungsbereichen wie beispielsweise kooperatives Arbeiten, computerunterstütztes Lernen oder Datenbankanwendungen einzusetzen.

Das World-Wide Web

Die meisten Informationen innnerhalb des World-Wide Web (WWW) werden dem Benutzer in Form von Dokumenten zugänglich gemacht, die gemäß den Konventionen der Hypertext Markup Language (HTML) erstellt wurden. Um einen Dokumentenaustausch zwischen heterogenen Plattformen zu ermöglichen, wurde mit HTML eine Dokumentenbeschreibungssprache gewählt, die nur die logische Struktur eines Dokuments wie Überschriften oder Aufzählungen mit Hilfe sogenannter Tags kenntzeichnet, aber keinerlei Layoutaspekte wie Schriftart oder Schriftgröße festlegt. Somit kann die Darstellung eines Dokuments jeweils an die verfügbare Infrastruktur angepaßt werden. Diese Dokumente, die neben Texten, auch Graphiken und Querverweise enthalten können, werden dem Benutzer mit Hilfe eines Frontends (Browsers) in graphischer Form präsentiert. Die Navigation entlang der Querverweise wird mit Hilfe eines Point-and-Click Interfaces realisiert und ermöglicht dem Benutzer eine einfache Interaktionsmöglichkeit. Eine Reihe von Graphikformaten, sowie Audio- und Videosequenzen können von den Browsern nicht direkt, sondern nur mit Hilfe von externen Programmen, sogenannten External Viewers bzw. Helper Applications, dargestellt werden.

Die Architektur des World-Wide Webs basiert auf dem Client-Server Paradigma, wobei sämtliche Ressourcen innerhalb des World-Wide Webs von den Servern verwaltet werden. Der Zugriff des Browsers auf HTML-Dokumente erfolgt unter Verwendung des Hypertext Transfer Protokolls (HTTP), das für die Dauer der Interaktion zwischen dem Client und dem Server eine logische Punkt-zu-Punkt Verbindung aufbaut. Die Adressierung der Ressourcen innerhalb des World-Wide Webs basiert auf einem physischen Namensschema, das jeder Ressource einen eindeutigen Namen, den Uniform Resource Locator (URL), zuordnet. Der URL einer Ressource setzt sich aus einem Protokollnamen, der Internetadresse des Servers und der Lokation der Ressource innerhalb des Servers zusammen. Die Aufnahme des Protokollnamens innerhalb des URLs ermöglicht neben dem Hypertext Transfer Protokoll die Verwendung anderer Protokolltypen und ermöglicht so die Integration bereits bestehenden Systemdienste wie ftp, telnet oder gopher innerhalb des World-Wide Webs. Zusätzlich zu dieser Funktionalität unterstützt das World-Wide Web mit Hilfe des Common Gateway Interfaces (CGI) auch den Zugriff auf externe Ressourcenverwalter wie beispielsweise Datenbanksysteme. Zu diesem Zweck startet der Server auf Anfrage des Clients lokal ein Program, das auch als CGI-Skript bezeichnet wird und in einer beliebigen Programmiersprache erstellt werden kann. Dieses CGI-Skript kann nun beliebige Aktionen ausführen und das Ergebnis dieser Aktionen als HTML-Dokument darstellen, das dann via WWW-Server zum Client gesendet und dort dargestellt wird. Zur Parameterübergabe an das CGI-Skript kann die HTML-Forms Schnittstelle verwendet werden, die ab HTML Version 2.0 verfügbar ist. Diese Schnittstelle stellt einige Basiselemente für graphische Benutzerschnittstellen wie Texteingabefelder, Buttons und Menüleisten zur Verfügung.

Der MediaStore-Browser

Zielsetzung

Die Zielsetzung des MediaStore-Browsers stellt die Realisierung eines World-Wide Web basierten Datenbank-Browsers dar, der in der Lage sein sollte einen Zugriff auf konventionelle textuelle und kontinuierliche Daten zu ermöglichen. Unter textuellen Daten werden im nachfolgenden Relationen im relationalen Modell bzw. Objekte im objektorientierten Modell verstanden. Kontinuierlichen Daten werden zeitabhängige Medien wie Video- oder Audiosequenzen zugerechnet. Um die Anwendbarkeit des Browsers für ein breites Anwendungsspektrum zu gewährleisten, wurde die Funktionalität der Benutzerschnittstelle unabhängig vom zugrundeliegenden Datenmodell (relational oder objektorientiert) realisiert. Bei der Implementierung wurde insbesonere auf die Verwendung von Standards wie ODBC (Open Database Communication) und CGI (Common Gateway Interface) geachtet.

Konzeption

Automatische Ermittlung des Datenbankschemas durch den Browser

Prinzipiell existieren zwei verschiedene Möglichkeiten, die Präsentation der Daten innerhalb des Datenbank-Browsers zu realisieren. Die erste Möglichkeit ist, die Präsentation der Daten vollständig dem Browser zu überlassen und die dazu notwendigen Schemainformationen zur Laufzeit zu ermitteln. Die zweite Möglichkeit ist, die Präsentation der Daten aufgrund einer externen Beschreibung durchzuführen. Diese Beschreibung beinhaltet dabei sowohl die Darstellung als auch die Abbildung von dargestellten Informationen auf Objekte der Datenbank. Falls die Präsentation der Daten vollständig dem Browser überlassen wird, besitzt diese Vorgehensweise den Nachteil, daß keinerlei Einfluß auf die Präsentation der Daten möglich ist. Der Vorteil dieser Vorgehensweise ist jedoch, daß das Auftreten von Inkonsistenzen mit der externen Beschreibung bei der Änderung des Datenbankschemas vermieden werden.

Hypertext-basierte Präsentation der Daten

Das Entwicklungziel der Benutzerschnittstelle war es eine möglichst intuitive Schnittstelle zum Datenbanksystem zu realisieren, die unabhängig vom zugrundeliegenden Datenmodell verwendet werden kann. Ein sehr wichtiger Gesichtspunkt beim Entwurf war die Realisierung einer Darstellung, die neben der Formulierung von Anfragen ebenfalls die Präsentation der Anfrageergebnisse ermöglicht. Auf diese Weise wird ein Dialog zwischen Benutzer und Datenbanksystem unterstützt, der dem Benutzer erlaubt seine Anfragen schrittweise zu verfeinern. Der Benutzer wird dadurch in die Lage versetzt Anfragen auf einem Objekttyp zu formulieren. Durch einfaches Anklicken können dabei diejenigen Attribute ausgewählt werden, die Teil des Ergebnisses sein sollen. Außerdem kann die Ergebnismenge durch die Eingabe von Attributenwerten zusätzlich noch eingeschränkt werden. Als standardmäßige Interaktion wurde eine tupelweise Darstellung der Ergebnisse gewählt, wobei aber bei Bedarf zu einer mengenorientierten Darstellung gewechselt werden kann. Beziehungen zwischen den Objekttypen, die im relationalen Datenmodell mittels Primärschlüssel-Fremdschlüssel Beziehungen und im objektorientierten Datenmodell mittels Verweise auf andere Objekte modelliert werden, werden innerhalb des Browsers als Hyperlinks dargestellt. Der Datenbank-Browser stellt somit eine Schnittstelle für den navigierenden Zugriff auf Datenobjekten zur Verfügung, die mit der eines Objektbrowsers für objektorientierte Systeme vergleichbar ist.

Präsentation von kontinuierlichen Daten unter Einhaltung der notwendigen Realzeitanforderungen

Auch das World-Wide Web bietet bereits die Möglichkeit an, Präsentationen von kontinuierlichen Daten mit Hilfe von externen Programmen durchzuführen. Die dabei verwendete Vorgehensweise ist jedoch für größere Datenmengen ungeeignet, da keine Überlappung zwischen Empfang und Präsentation der Daten stattfindet, sondern sämtliche Daten zuerst vollständig übertragen und anschließend dargestellt werden.

Für den Benutzer hat diese Entkopplung von Empfang und Darstellung zur Folge, daß er gezwungen ist die gesamte Dauer der Datenübertragung abzuwarten, auch wenn er nur an einem kleinen Ausschnitt der Daten interessiert ist. Aus diesem Grund stellt die Überlappung von Darstellung und Empfang bei der Präsentation kontinuierlicher Daten ein wichtiger Entwurfsgrundsatz für das Gesamtsystem dar.

Eine Überlappung von Darstellung und Empfang bei der Präsentation erfordert jedoch die Einhaltung von Realzeitanforderungen um Qualitätsverluste in Form von Gleichlaufschwankungen zu vermeiden. Eine Einhaltung von Realzeitanforderungen bzw. einer für die Präsentation notwendigen Dienstgüte (Quality of Service) kann nur bei einer Reservierung sämtlicher Ressourcen wie Prozessorleistung, Netzwerkbandbreite oder Hauptspeicherbedarf entlang des Datenpfades gewährleistet werden. Um diese Reservierung vornehmen zu können, müssen sämtliche Ressourcen über ein entsprechendes Reservierungsprotokoll verfügen, eine Situation, wie sie in heutigen Betriebssystemen nicht anzutreffen ist. Entsprechende Dienste sind bisher nur in Forschungsprototypen realisiert und auch ATM-Netzwerke, die eine Bandbreitenreservierung erlauben, stellen nicht nur innerhalb von Weitverkehrsnetzwerken, wie dem World-Wide Webs noch die Ausnahme dar. Obwohl eine Gewährleistung der Dienstgüte für die gesamte Übertragungsdauer und den gesamten Datenpfad nicht garantiert werden kann, ist der Einsatz einer lokalen Komponente auf Datenbankseite durchaus sinnvoll. Diese Komponente überprüft beim Eintreffen einer neuen Anforderung auf die kontinuierlichen Daten, unter Berücksichtigung der augenblicklichen Auslastung des Systems, ob die Dienstgüte zumindest lokal gewährleistet werden kann. Diese Komponente, die auch als Reservierungsmanager bezeichnet wird, entscheidet aufgrund der verfügbaren E/A-Bandbreite und der existierenden Netzwerkbandbreite, ob eine Gewährleistung einer neuen Anforderung, die Beeinträchtigung von aktiven Anforderungen zur Folge hätte und zurückgewiesen werden sollte, oder gewährleistet werden kann (Admission Control).

Durch den Einsatz eines Feedback-Mechanismus bei der Präsentation, der dem Reservierungsmanager Verletzungen der Dienstgüte aufgrund von temporären Überlastsituationen mitteilt, kann der Reservierungsmanager mit Hilfe von Statusinformationen, wie z.B. der Anzahl der zu spät eingetroffenen Datenpakete bzw. der momentan erzielten Datenrate bei der Präsentation, die Dienstgüte kontinuierlich erniedrigen (Graceful Degredation) bis keine Verletzungen der Dienstgüte mehr festgestellt werden kann. Dieser Mechanismus ermöglicht auch bei geringerer Dienstgüte eine für den Benutzer akzeptable Form der Präsentation bereitzustellen.

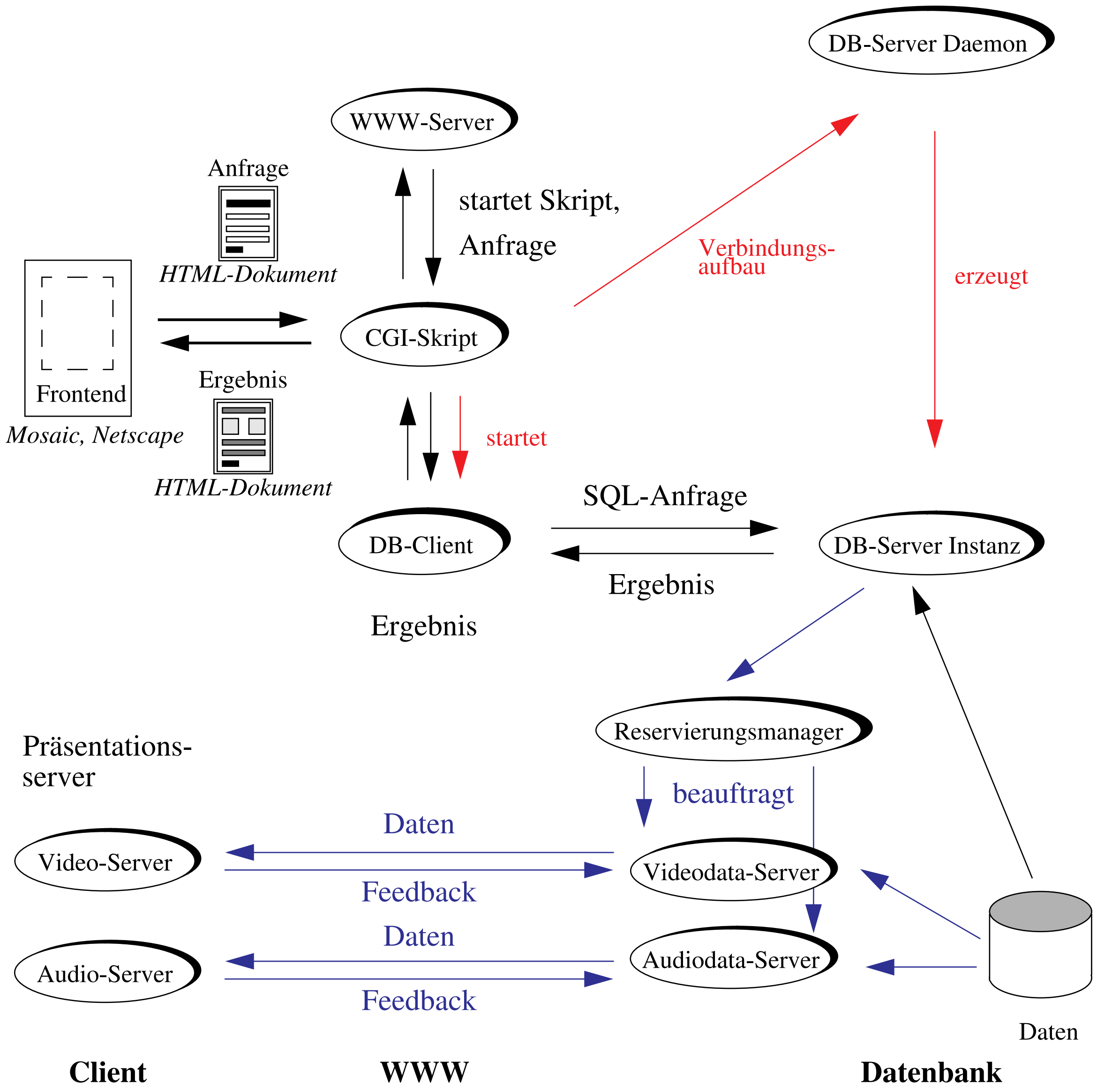
Effiziente Interaktion zwischen World-Wide Web und Datenbanksystem

Die Anbindung von externen Ressourcenverwaltern wie Datenbanksysteme wird innerhalb des World-Wide Webs mittels CGI-Skripten ermöglicht, so daß die Datenbankzugriffe bereits innerhalb der CGI-Skripte realisiert werden könnten. Diese Vorgehensweise besitzt allerdings den Nachteil, daß für jeden Datenbankzugriff eine Verbindung zum Datenbanksystem aufgebaut werden muß und daher sehr ineffizient ist. Durch die Einführung einer zusätzlichen Komponente, die sämtliche Zugriffe auf die Datenbank durchführt und im folgenden als DB-Server Instanz bezeichnet wird, kann eine dauerhafte Verbindung zur Datenbank realisiert werden, die für die gesamte Dauer der Interaktion mit der Datenbank verwendet werden kann. In dieser Architektur führt das CGI-Skript die Abbildung der Anfrage des Benutzers in SQL durch, sendet diese SQL-Anweisung an die DB-Server Instanz, empfängt die Anfrageergebnisse und bereitet sie als HTML-Dokument auf. Zur Realisierung von tupelorientierten Zugriffen müßten Zwischenergebnisse und Statusinformationen zwischen aufeinanderfolgenden Anfragen innerhalb der DB-Server Instanz gespeichert werden, die eine erhöhte Ressourcenbelastung des Datenbanksystems zur Folge hätte. Aus diesem Grund ist die Einführung einer weiteren Komponente, dem DB-Client, zur Speicherung der Zwischenergebnisse sinnvoll. Auf diese Weise kann die Ressourcenbelastung des Datenbanksystems verringert und eine Erhöhung des Durchsatzes erzielt werden.

Architektur

Die Architektur des Datenbank-Browsers läßt sich direkt aus der Beschreibung der notwendigen Komponenten ableiten, wie sie innerhalb der Konzeption im vorhergehenden Abschnitt beschrieben wurden. Die

Beschreibung der Architektur im folgenden Abschnitt erfolgt aufgrund ihrer Lokation innerhalb des Gesamtsystems.



Auf der Seite des Benutzers sind das *World-Wide Web Frontend*, das mittels HTML-Forms die Benutzerschnittstelle des Datenbank-Browsers realisiert, und die *Präsentationsserver für kontinuierliche Daten* angesiedelt. Diese Server sind neben der Präsentation von Audio -und Videosequenzen auch für die Überwachung der Qualitätsanforderungen zuständig und melden eventuelle Verletzungen der Dienstgüte an die entsprechenden Server auf Datenbankseite zurück, um eine Anpassung der Dienstgüte zu erreichen.

Der World-Wide Web basierte Teil des Systems wird aus dem *World-Wide Web Server* selbst, der im wesentlichen für die Kommunikation mit dem Frontend und den Start des CGI-Skripts zuständig ist, dem CGI-Skript und dem DB-Client gebildet. Nach dem Login des Benutzers ist das *CGI-Skript* für den Verbindungsaufbau zur Datenbank und für den Start des DB-Clients zuständig. Bei Beendigung der Interaktion zwischen Datenbank und Benutzer initiiert das CGI-Skript ebenfalls wieder den Verbindungsabbau. Ansonsten dient das CGI-Skript lediglich als Kommunikationskanal zwischen Benutzer und DB-Client. Es übermittelt dem DB-Client die Anfrage des Benutzers und erhält als Ergebnis ein HTML-Dokument, das anschließend via

WWW-Server an den Browser gesendet und dort dargestellt wird. Der *DB-Client* übersetzt die Benutzeranfrage in eine SQL-Anweisung, die dann an die für ihn zuständige DB-Server Instanz gesendet wird. Als Antwort erhält er die Anfrageergebnisse, die er, unter Berücksichtigung des zugrundeliegenden Datenbankschemas, als HTML-Dokument aufbereitet.

Auf Datenbankseite existieren der *DB-Server Daemon*, der auf Anfrage des CGI-Skripts für jeden neuen Benutzer eine neue DB-Server Instanz startet, der Reservierungsmanager, der für die Gewährleistung der Dienstgüte zuständig ist und die Daten-Server für kontinuierliche Daten. Die *DB-Server Instanz* führt auf Anfrage des DB-Clients die eigentlichen Datenbankzugriffe durch und sendet das Ergebnis zurück an den DB-Client. Beinhaltet die Anfrage ebenfalls Zugriffe auf kontinuierliche Daten wird der *Reservierungsmanager* kontaktiert, der dann, aufgrund der augenblicklichen Systemauslastung, die Entscheidung über die Gewährleistung des Zugriffs durchführt. Falls der Zugriff gewährleistet wird, startet er die entsprechenden *Daten-Server*, die dann die kontinuierliche Daten direkt zu den Präsentationsservern senden. Kann der Zugriff auf die kontinuierlichen Daten nicht gewährt werden, sendet er eine entsprechende Mitteilung an die DB-Server Instanz, der diese Mitteilung als Teil der Ergebnisse an den DB-Client weitergibt.

Stand der Implementierung

Die aktuelle Implementierung umfaßt mit Ausnahme des Basisdienstes zur Ressourcenreservierung (Reservierungsmanager) sämtliche Komponenten, die innerhalb der Konzeption vorgestellt wurden. Die Benutzerschnittstelle des Datenbank-Browsers wurde unter Verwendung von HTML-Forms realisiert, wobei der Netscape Browser Version 2.0 als Frontend zum Einsatz kommt. Als World-Wide Web Server wird der HTTP-Server von NCSA verwendet, alle anderen Komponenten wurden in der Programmiersprache C implementiert. Die Datenbankzugriffe auf textuelle Daten erfolgen mittels ODBC-Schnittstelle von der DB-Server Instanz auf das Datenbanksystem DB2/6000 V2.1 von IBM. Aufgrund der Verwendung der ODBC-Schnittstelle ist die Einbindung von Datenbanksystemen anderer Hersteller jederzeit möglich. Die Zugriffe auf die kontinuierlichen Daten, die in Form von BLOBs (Binary Large Objects) ebenfalls in der Datenbank gespeichert werden, erfolgt unter Verwendung der Embedded-SQL Schnittstelle. Da der Datentyp BLOB und die darauf definierten Zugriffsoperationen nicht Teil des aktuellen SQL Standards (SQL2) sind, existieren herstellerabhängige Implementierungen, so daß dieser Teil der Architektur an das Datenbanksystems des jeweiligen Herstellers angepaßt werden muß. Die derzeit verwendeten Präsentationsserver beschränken sich auf den MPEG-1 Video Player V2.1 der Universität Berkeley, der ebenso wie der MPEG Audio Player V1.1 der TU Berlin, als Public Domain Software verfügbar ist und ohne zusätzliche Hardware auf einer Vielzahl von Plattformen eingesetzt werden kann. Beide Server wurden entsprechend modifiziert, um eine Netzwerkanbindung zu ermöglichen. Die Übertragung der Daten zwischen den einzelnen Komponenten erfolgt mittels TCP/IP-Protokoll.

Ausblick

Ein vorrangiges Ziel bei der weiteren Vorgehensweise, stellt die Integration eines Basisdienstes für die Ressourcenreservierung in das Gesamtsystem dar. Daran anschließend soll eine Portierung der Präsentationsserver unter Verwendung der Programmiersprache Java durchgeführt werden, so daß diese als Java-Applikation (Applets) innerhalb des Browsers ablaufen und die Präsentation der Daten innerhalb eines HTML-Dokuments erfolgen kann. Durch den Einsatz der Programmiersprache Java innerhalb des Browsers ist es ebenfalls möglich zwischen dem Java Applet, das die Benutzerschnittstelle implementiert, und dem DB-Client eine direkte Kommunikation zu realisieren ohne dazu ein CGI-Skript verwenden zu müssen. Es ist anzunehmen, daß insbesondere diese letzte Erweiterung zu einer erheblichen Effizienzsteigerung des Gesamtsystems beiträgt.

Kontaktadresse

Dipl.-Inform. Peter Kutschera
Universität Stuttgart
Institut für Parallele und Verteilte Höchstleistungsrechner (IPVR)
Breitwiesenstaße 20 - 22
70565 Stuttgart
Telefon 0711 / 7816 - 349
Telefax 0711 / 7816 - 424
Email Peter.Kutschera@informatik.uni-stuttgart.de