

Der Smart Data Server

Neue Konzepte und praktische Anwendung einer Middleware-Architektur

Dissertation

Dipl.-Inform. Uwe Roth

**Fachbereich IV
Informatik
Universität Trier**

Oktober 2003

Zusammenfassung

Die Geschichte verteilter Systeme und Anwendungen hat die unterschiedlichsten Technologien hervorgebracht: Client-Server-Architekturen, Peer-To-Peer-Netzwerke und Komponentensysteme sind nur einige Vertreter. Die vorliegende Arbeit ist im Bereich der Middleware-Architekturen angesiedelt, einem zur Zeit sehr stark beachteten Zweig der verteilten Anwendungen. Als Mittler zwischen den Anwendungen auf der einen Seite sowie Datenbanken, eMail-Systemen oder weiteren Servern auf der anderen Seite, schlägt sie die Brücke zu heterogenen IT-Landschaften. Sie verbirgt Details und Änderungen dieser IT-Umgebungen und schafft gleichzeitig einen transparenten Zugriff auf sie.

Die Forschung hat viele Technologien im Middleware-Umfeld hervorgebracht und setzt bei den Zielen unterschiedliche Schwerpunkte. Eine Sicht auf die Middleware hat den Entwickler im Fokus und soll ihn bei der Erstellung von Anwendungen und Lösungen unterstützen. Aus diesem Grund stehen hier die Schnittstellen zum Server und zur Server-Infrastruktur sowie die Einbettung der Dienste in die Middleware im Vordergrund. Der interne Aufbau sowie die inneren Datenflüsse der Middleware dienen nur der Erfüllung dieser Ziele.

Eine andere Sichtweise stellt den inneren Aufbau in den Fokus der Forschung. Hier ist das Ziel die Schaffung von flexiblen und erweiterbaren Server-Strukturen sowie die Effizienz von internen Abläufen und Datenflüssen. Damit ist eine einfache Anpassung an verschiedene Einsatzumgebungen sowie die Fähigkeit, auf die Leistungsmerkmale von Clients individuell eingehen zu können, möglich.

Die im Rahmen dieser Arbeit entstandene Middleware-Architektur „Smart Data Server (SDS)“ setzt Konzepte um, die beide Sichtweisen miteinander kombiniert.

Ein Schwerpunkt neben der Entwicklung neuer Konzepte, liegt in der praktischen Anwendbarkeit der entwickelten Middleware. Die vorliegende Arbeit zeigt, dass der Smart Data Server die Praxistauglichkeit seiner Konzepte unter realen Bedingungen unter Beweis stellen kann und zeigt gleichzeitig, welche Konstrukte für den praktischen Einsatz tatsächlich notwendig sind.

Ein zweiter Schwerpunkt liegt in der Entwicklung von Mechanismen zur Abarbeitung von streamingfähigen Anfragen. Üblicherweise folgt auf die Übermittlung von Anfragen an einen Server zunächst die Berechnung einer Antwort und anschließend deren Rücktransport. Bei Streaming-Anfragen ist neben der Übermittlung von Anfrageparametern ein kontinuierlicher und in seiner Dimension nicht beschränkter Datenstrom Bestandteil der Anfrage. Der Server startet seine Berechnungen wenn möglich schon vor dem Ende der Datenübermittlung und liefert schon bei Vorliegen von Teilergebnissen diese an den Client. Der benötigte Bedarf an Speicherplatz im Server kann so optimiert werden, da nur so viele Daten der Anfrage im Server vorgehalten werden müssen wie zur Berechnung tatsächlich notwendig sind und nicht etwa die gesamte Anfrage.

Die Entwicklung eines Streaming-Mechanismuses muss entsprechende Möglichkeiten in der Anfragestruktur sowie in der Architektur der Middleware vorsehen. Der Smart Data Server schafft mit der Einführung einer streamingfähigen Anfragestruktur die Voraussetzung für die Verarbeitung solcher Anfragen. Zusätzlich muss eine Server-Architektur geschaffen werden, die das Abarbeiten von Streaming-basierten Anfragen unterstützt.

Hier liegt der dritte Schwerpunkt der Arbeit: die Schaffung einer flexiblen und erweiterbaren Server-Struktur sowie die Effizienz von internen Abläufen und Datenflüssen unter Berücksichtigung der Anforderung einer streamingfähigen Anfragebearbeitung. Diese Anforderungen werden mit der neu entwickelten Technologie der inneren Workflow-Programme erfüllt. Innere Workflow-Programme repräsentieren Netzwerke von hochgradig unabhängigen Serverkomponenten. Durch die Umgestaltung dieser Netze sind neue Serverstrukturen möglich, um neuen Einsatzbedingungen angepasst werden zu können. Neben diesem statischen Netzwerk von Server-Komponenten kann zur Laufzeit eine Pipeline zur Abarbeitung von Datenströmen aufgebaut werden, mit der dann letztendlich die Streamingfähigkeit des Servers hergestellt wird.