



Darf's ein bisschen schneller sein?

Trotz großem Hauptspeicher, schnellen Prozessoren und Breitbandverbindungen leiden viele Anwendungen unter einer schlechten Performance. Die „Heilung“ ist oftmals ganz einfach und kann eine Anwendung um den Faktor 100 und mehr beschleunigen.

Text: Romana Pockstaller
Fotos: iStockphoto, Romana Pockstaller

Das Problem ist in Firmen, die komplexe Software nutzen, hinlänglich bekannt: Aufwändige Verarbeitungen werden immer wieder wegen Timeouts vorzeitig beendet oder Internetseiten bauen sich nur schleppend auf. Der Mitarbeiter verbringt einen Gutteil seiner Arbeitszeit mit Warten – für ihn frustrierend, für den Arbeitgeber teuer. Die hektische Suche nach Leistungsgipfeln und deren Beseitigung beginnt. Dann werden in der Regel Analysewerkzeuge gestartet, Server aufgerüstet, Datenbanksoftware aktualisiert und Programmteile optimiert.

Doch das Programm wird trotzdem nicht wesentlich schneller.

Neu betrachtet – neu gedacht

Der Kern des Geschwindigkeitsproblems liegt nämlich häufig woanders. Denn obwohl die Anwendungen von Softwarearchitekten bereits mit einem sauberen Design versehen worden sind, kann das Programm durch einen Denk- oder Architekt-

turfehler komplett an der eigentlichen Verwendung vorbei gehen. Da hilft dann meist nur ein frisches Paar Augen, um den Überblick wieder zu bekommen, eine neue Sicht fürs Wesentliche zu entwickeln und so den Weg für einen alternativen Lösungsansatz zu eröffnen.

Hannes Reisinger von HaRe Software & Beratung hat sich unter anderem der Effizienzsteigerung von Enterprise-Anwendungen verschrieben. Seine 20-jährige Erfahrung in der Branche bei Großkunden wie Hypo-Vereinsbank München, Hewlett Packard in Wien oder Infineon kombiniert mit einer großen Offenheit für Neues lassen ihn bei der Analyse und Lösung von Performance-Problemen aus einem großen Fundus an Wissen, Erfahrung und Innovationsfreude schöpfen. Oft schon sind ihm Anwendungen untergekommen, bei denen „mit Kanonen auf Spatzen geschossen wird“, wie er sagt. Dies führe dazu, dass riesige Mengen an Daten auslastungsintensiv und zeitaufwendig die Runde drehen müssen, bevor das Programm die

gewünschte Funktion ausführen kann. Oft wären Umwege dieser Art gar nicht notwendig, ist Reisinger überzeugt. Sie ergeben sich gern daraus, dass sich die Anforderungen an ein Programm fast immer entweder bereits während der Programmierung bzw. später in der Anwendung verändern, ohne dass darauf von den Softwarearchitekten entsprechend eingegangen wird. „Man fährt ja auch nicht mit der Daunenjacke in die Südsee, nur weil man sich ursprünglich mal ein Prospekt für Grönland geholt hat“, meint Reisinger lakonisch.

Schön (und | aber) langsam

Unzählige Beispiele wie das folgende sind dem erfahrenen Entwickler im Laufe der Jahre schon untergekommen: Eine Anwendung bekommt stündlich Daten von einem Großrechner in Form einer Textdatei geliefert und muss diese Daten dann bei Bedarf für verschiedenste Anfragen aus dem Internet in XML bereitstellen. Die Lösung des Softwarearchitekten beinhaltet einen eigenen J2EE-Anwendungsserver, der die Daten der Textdatei für die Datenbank optimiert unwandelt und speichert. Bei den Anfragen werden die Daten dann wieder aus der Datenbank geholt und aufwändig aufbereitet. Aus Sicht des Softwarearchitekten eine saubere Lösung, mit der er sicher zufrieden war. Für den Anwender aber brachte die Umwandlung und Ablage sowie das Auslesen und Aufbereiten der Daten nur Nachteile: Die Anwendung war extrem langsam, während der Verarbeitung der Textdatei wurde sehr viel Speicher verbraucht und bei gleichzeitigen Abfragen stürzte das Programm regelmäßig ab.

Einfach (und | aber) effizient

„Mein Ziel ist es, die Anwendung für den User mit wenig Aufwand möglichst schnell und effizient zu machen und dies bei geringem Ressourcenverbrauch“, so Reisinger, dem es im beschriebenen Fall gelungen ist, die Anwendung um mehr als das Hundertfache zu beschleunigen. Das bedeutet, dass die Abfrage einer Internetseite, die vor der Überarbeitung zwei Minuten dauerte, jetzt in weniger als einer Sekunde beantwortet wird.

Die von Reisinger umgesetzte Lösung ist deshalb so effizient, weil sie sich auf die wesentlichen Aufgaben konzentriert und auf überflüssige Zwischenschritte wie das Umwandeln, Speichern und wieder Holen der Daten verzichtet. Stattdessen werden aus der Textdatei unmittelbar Bausteine für die Antworten zukünftiger Anfragen generiert und im Speicher behalten. So liegt die Lösung schon bereit und braucht nur noch abgerufen zu werden.

Das Ganze wurde auf Kundenwunsch in Java realisiert, braucht nur einen Bruchteil des Hauptspeichers und kaum CPU-Zeit, keinen Datenbankserver und die Anfragen werden extrem schnell beantwortet. Aufgrund der Einfachheit ist die Ausfallsicherheit durch einen zweiten Server mit der gleichen Textdatei gegeben. Wäre die Anwendung nicht jetzt schon so unheimlich schnell, dann könnte sie problemlos auf mehreren Servern betrie-

ben werden. Weitere Vorteile sind: weniger Quellcode ist leichter zu pflegen und die Wartung eines simplen Java-Dienstes ist sehr viel einfacher als die eines J2EE- und eines DB-Servers.

„Alles Wahre ist einfach“

... sagt Karl Heinrich Waggerl und diese Erkenntnis ist auch Teil der Unternehmensphilosophie von HaRe Software & Beratung. „Die große Kunst ist es, aus der Flut von Informationen über Datenmodell und Detailprobleme die eigentliche Anforderung herauszufinden, die dann unter Umständen mit vergleichsweise einfachen Mitteln erfüllt werden kann“, sagt Reisinger. „Dazu braucht es viel Erfahrung in der Softwareentwicklung und Offenheit für Neuerungen“, ist der passionierte Programmierer überzeugt.

Radikale Verbesserungen erfordern aber auch aufgeschlossene Kunden, die bereit sind, sich auf Neues einzulassen und glauben können, dass gute Lösungen auch einfach sein dürfen. ●



Programme aus einem neuen Blickwinkel betrachten, um effizienter zu arbeiten, ist eines der Hauptanliegen von Hannes Reisinger, Geschäftsführer von HaRe Software und Beratung.

Alles auf Schiene mit „Ruby on Rails“

Als vielversprechende und effiziente Alternative in der Webentwicklung stellt sich das Werkzeug „Ruby on Rails“ dar. Damit lassen sich sehr schnell Webanwendungen entwickeln, die anderen, wesentlich aufwändiger programmierten, in nichts nachstehen. Im Gegenteil: Ruby on Rails kann mit vielen Vorteilen punkten, die sowohl dem Unternehmen als auch dem Entwickler entgegenkommen.

Weniger ist mehr – und schneller

Mit Ruby on Rails kann die gleiche Aufgabenstellung mit einem Bruchteil von Quellcodezeilen realisiert werden als in Java. Ermöglicht wird dies u.a. durch den Grundsatz „don't repeat yourself“. Ein weiterer entscheidender Faktor ist die kurze Einlernphase. Schon nach einer Woche Schulung in Ruby on Rails kann ein Entwickler, der beispielsweise mit Java vertraut ist, produktiv Webanwendungen entwickeln. Dies motiviert den Mitarbeiter und spart dem Unternehmen bares Geld. Wird derselbe Mitarbeiter z.B. auf Java Enterprise Edition geschult, müssen mehrere Wochen veranschlagt werden, bis er wesentlich unproduktiver Anwendungen fürs Web programmieren kann.

Im Hätetest erprobt

Das bekannteste Vorzeigebeispiel für eine erfolgreiche Webanwendung basierend auf Ruby on Rails ist die Website von Twitter, welche mit rund 10.000 Abfragen pro Sekunde fertig werden muss. Dies sollte auch Zweifler von der Praxistauglichkeit von Ruby on Rails für Anwendungen aller Größen überzeugen. Und zu guter Letzt bietet dieses Werkzeug eine echte Chance für die Entwicklungsabteilungen, auch in Zeiten sinkender Budgets gleichbleibende oder sogar steigende Anforderungen in Zukunft zu meistern!