

"Die Logik gehört dahin, wo die Daten sind"

› Joe DeSantis, Vice President of Software Development von InterSystems, über den Markt für Objektdatenbanken im Allgemeinen und die Rolle von Caché darin im Speziellen.

(16.12.2009 10:00:00)

CW: Als InterSystems 1997 Caché vorstellte, verkündeten Experten gerade den Siegeszug der Objektdatenbanken. Warum ist daraus nichts geworden?



Joe DeSantis, Vice President of Software Development von InterSystems.

DeSantis: Die glänzende Zukunft der Objektdatenbanken war eigentlich schon viel früher prophezeit worden. Als wir [Caché](#) herausbrachten, war uns bereits klar, dass reine Objektdatenbanken der ersten Generation nie erfolgreich sein würden, weil diese sich kaum für kommerzielle Anwendungen eigneten. Man konnte keine [Standard-SQL-Reports](#) erstellen, genauso wenig waren sie für große Anwenderzahlen oder hohen Transaktionsdurchsatz geeignet. Für ganz große Verunsicherung im Objektdatenbankmarkt sorgten schließlich SQL 99 und andere Ansätze für neue Standards, die noch weniger erfolgreich waren als die Objektdatenbanken selbst.

Zum Durchbruch verhalf Caché die Idee, eine Objektdatenbank zu schaffen, die zugleich auch eine relationale Datenbank war. So hatten wir einerseits für diejenigen etwas zu bieten, die eine Objektdatenbank wollten, andererseits gab es einen Evolutionspfad, wenn jemand ganz einfach relational beginnen und dann mit Objekten weitermachen wollte. Auch wenn die Objektdatenbanken nicht den Marktanteil erobert haben,

der von einigen vorhergesagt wurde, sind wir sehr zufrieden mit dem, was wir erreicht haben.

CW: Seit einigen Jahren ist Caché nicht mehr das einzige Produkt von InterSystems. Inzwischen gibt es eine ganze Produktpalette inklusive einer Integrationsumgebung. Zählen Datenbanken für Sie überhaupt noch zum strategischen Kerngeschäft?

DeSantis: Nach wie vor bildet Caché die Basis aller unserer Aktivitäten. Die Integrations- und Entwicklungsplattform [Ensemble⁵](#) setzt auf dieser Datenbanktechnik auf. Messages sind keine [relationalen Tabellen⁶](#) - sie besitzen eine Struktur, die Sie in der Regel auch erhalten möchten, weil sie eine Bedeutung hat. Außerdem binden wir die Kunden in die Entwicklungsstufen unserer Technik mit ein. Hätten wir Caché nur intern genutzt und niemand anderen dran gelassen, wäre die Datenbank nie so leistungsstark und funktionsreich geworden wie in der Zusammenarbeit mit Partnern, die tagtäglich neue Anwendungsfälle finden, an die wir nie gedacht hätten.

Ein riesiger Schritt war für uns zum Beispiel die Möglichkeit, [XML⁷](#) in der Datenbank zu speichern. Viele meinten, dass nach den Objektdatenbanken XML-Datenbanken der nächste große Wurf ist. Auch das ist ja nie eingetreten. Aber dass wir Caché mit nativem XML-Support ausgestattet haben, hatte enorme Konsequenzen. Das gilt für das, was unsere Kunden damit anfangen können, aber auch für unsere eigene Entwicklung - wir nutzen die XML-Fähigkeiten bei nahezu allem, was wir in unseren Produkten machen.

CW: Ist die native Speicherung von Objekt- und XML-Daten wirklich noch ein Vorteil? Heute beherrschen doch die meisten Entwicklungsumgebungen, Datenbanktreiber und Frameworks wie .NET von Hause aus objektrationales Mapping?

DeSantis: Das hängt davon ab, wie schnell Sie Ergebnisse sehen möchten. Anstatt viel Zeit in den Einsatz von Mapping-Tools zu investieren, möchten wir, dass dies automatisch erledigt wird. Wenn Entwickler Spaß daran haben, ihren [Java¹⁰](#)-Klassen [Oracle¹¹](#)-Tabellen zuzuweisen und dies bei jeder Feature-Änderung zu wiederholen, wenn sie stets alle Daten entladen und wieder importieren müssen und es nicht schlimm finden, dafür Monate anstatt ein paar Tage aufzubringen, dann ist das okay. Probleme durch immer mehr Tools zu beheben, anstatt sie grundlegend zu lösen, ist aber für uns keine vernünftige Alternative.

CW: Was sticht einem Entwickler beim Arbeiten mit Caché im Vergleich zu einer typischen relationalen Datenbank sofort ins Auge?

DeSantis: Was uns grundlegend von anderen unterscheidet, ist die feste Überzeugung, dass die Logik dahin gehört, wo die Daten sind. Vor vielleicht 20 Jahren hat die Branche einen Paradigmenwechsel zu [Three-Tier-Architekturen¹²](#) vollzogen. Das Konzept war, dass die Datenbank alle Daten an die mittlere Schicht liefern sollte, wo die gesamte Logik enthalten ist. Von der Architektur her gesehen, ist das ein sinnvolles Konzept, aber in der Praxis bringt es viele Probleme mit sich. Selbst heute befinden sich die meisten Daten nicht in relationalen Datenbanken sondern in Betriebssystemdateien, Tabellenkalkulationsblättern, auf dem Mainframe, an allen möglichen Orten. Dass die Logik so weit von den Daten entfernt ist, hat dazu geführt, dass inzwischen immer mehr in [Caching¹³](#)-Techniken investiert wird. Anstatt sich auf die beauftragten Lösungen zu konzentrieren, verbringen heutige Anwendungsprogrammierer einen Großteil ihrer Zeit mit Überlegungen, wie sie ihrem System ausreichend Performance abtrotzen können. Eigentlich sollten das die Hersteller für sie tun.

Der Hauptunterschied bei der Programmierung mit Caché besteht darin, dass Sie auf die benötigten Daten in einer Vielzahl an Formaten zugreifen können. Je nach Bedarf können wir Daten als Objekte öffnen, SQL-Abfragen darauf ausführen, sie in XML darstellen und ganz wichtig, wir können direkt als multidimensionale Arrays mit ihnen arbeiten. Weil die Daten wirklich nah an der Logik liegen, können die Programmiererteams bei Caché kleiner ausfallen als bei anderen Lösungen.

CW: War es nicht das Hauptziel aller Objektdatenbanken, die Logik nahe bei den Daten zu halten?

DeSantis: Dieses Konzept haben wir in der Tat mit den reinen Objektdatenbanken gemeinsam. Wir unterscheiden uns aber in der höheren Flexibilität aufgrund zusätzlicher Optionen. Alle Daten als verbundene Objekte anzuzeigen, ist nur eine der Möglichkeiten. Vor allem bei Transaktionen lässt sich mit Objekten viel natürlicher arbeiten als mit SQL. Wenn ich jedoch alle Mitarbeiter mit einer bestimmten Eigenschaft finden möchte, ist das nicht mehr so einfach, wenn ich dazu alle meine Objekte durchforsten, jedes öffnen und einzeln untersuchen muss. Unser Vorteil ist, dass wir zweigleisig fahren: Alle Daten als Objekte zu sehen, ebenso gut aber auch jederzeit Abfragen darüber laufen lassen zu können.

CW: Wie überzeugen Sie Anwender, die bereits gängige Datenbankprodukte im Einsatz haben, zusätzlich noch Caché zu kaufen?

DeSantis: Es gibt in der Regel zwei Möglichkeiten, wie wir neue Kunden gewinnen. Viele unserer Partner verfügen über Lösungen, in denen Caché eingebettet als Teil der Anwendung mitgeliefert wird. Kunden nehmen dies oft gar nicht wahr.

Die andere typische Situation für uns ins Spiel zu kommen, ist, wenn eine grundsätzliche Aufgabe, die bislang mit selbst gestrickten Tools erledigt wurde, nun besser gelöst werden soll. Zum Beispiel haben viele der großen Finanzdienstleister ihre eigenen In-Memory-Datenbanksysteme im Einsatz. Denn auch wenn sie unternehmensweit vielleicht mit Sybase oder Oracle arbeiten, keine dieser Datenbanken ist schnell genug, um Finanzdaten in Echtzeit zu verarbeiten. So wird alles in selbst gezimmerten In-Memory-Systemen verarbeitet und erst am Ende des Tages in den relationalen Datenbanken geparkt. Mit einigen der speziellen Anbindungen von Caché lässt sich dieselbe Leistung wie mit einer solchen In-Memory-Lösung erzielen. Hinzu kommen dann aber unter anderem die Backup- und Failover-Features einer echten Datenbank. Manchmal ist es irrwitzig zu sehen, wie viele Unternehmen auf mögliche Wettbewerbsvorteile verzichten, nur weil sie glauben, bei den Datenbanken sei die Spitze der Entwicklung vor 20 Jahren erreicht worden und die Produkte seither austauschbar.

- 1 <http://www.intersystems.de/>
- 2 <http://de.wikipedia.org/wiki/Objektdatenbank>
- 3 <http://www.intersystems.de/cache/index.html>
- 4 <http://de.wikipedia.org/wiki/SQL>
- 5 <http://www.intersystems.de/ensemble/index.html>
- 6 http://de.wikipedia.org/wiki/Relationale_Datenbank
- 7 http://de.wikipedia.org/wiki/Extensible_Markup_Language
- 8 <http://msdn.microsoft.com/de-de/netframework/default.aspx>
- 9 <http://www.it-visions.de/glossar/alle/838/lexikon.aspx>
- 10 <http://www.java.com/de/download/manual.jsp>
- 11 <http://www.oracle.com/global/de/index.html>
- 12 <http://de.wikipedia.org/wiki/Schichtenarchitektur>
- 13 <http://de.wikipedia.org/wiki/Cache>

IDG Business Media GmbH

Alle Rechte vorbehalten. Jegliche Vervielfältigung oder Weiterverbreitung in jedem Medium in Teilen oder als Ganzes bedarf der schriftlichen Zustimmung der IDG Business Media GmbH. DPA-Texte und Bilder sind urheberrechtlich geschützt und dürfen weder reproduziert noch wiederverwendet oder für gewerbliche Zwecke verwendet werden. Für den Fall, dass in Computerwoche unzutreffende Informationen veröffentlicht oder in Programmen oder Datenbanken Fehler enthalten sein sollten, kommt eine Haftung nur bei grober Fahrlässigkeit des Verlages oder seiner Mitarbeiter in Betracht. Die Redaktion übernimmt keine Haftung für unverlangt eingesandte Manuskripte, Fotos und Illustrationen. Für Inhalte externer Seiten, auf die von Computerwoche aus gelinkt wird, übernimmt die IDG Business Media GmbH keine Verantwortung.