

Embedded-Datenbanken

# Pflegeleicht

**Bei kleinen Embedded Systemen war es lange üblich, die Datenverwaltung einfach mit dem Rest der Software selbst zu programmieren. Seit »Embedded« immer komplexer wird, ist das keine sinnvolle Vorgehensweise mehr. Beim Rückgriff auf Standardlösungen ist jedoch zu beachten, dass nicht jede Datenbank auch für Embedded Systeme geeignet ist.**

---

Leon Houwen

---

Wie so oft gibt es auch bei diesem Thema eine Begriffsverwirrung. Der Begriff »embedded Datenbank« wurde um 1980 eingeführt und beschreibt ein DBMS (Datenbank-Management-System), das in die Applikation – im Unterschied zu Mainframe-Datenbanken – eingebunden wurde. In dieser Zeit hatte ein DBMS nichts mit einem Embedded System gemein. Heute verfügen eingebettete Systeme häufig über 32-Bit-Prozessoren, haben mehrere Megabyte Speicher und arbeiten mit Echtzeit- oder anderen Betriebssystemen. Anwendungen in der Unterhaltungselektronik, der Automatisierungstechnik sowie in der Kommunikation erfordern eine effiziente Verwaltung der anfallenden Daten. Datenspeicherung und -abfrage erfordern daher ein DBMS. Viele Hersteller von Datenbanken konzentrieren sich heute auch auf den Embedded-Markt, wobei je-

doch nicht jede Lösung tatsächlich auch für eingebettete Systeme geeignet ist. Die Programmierschnittstelle (API) einer Datenbank hat Auswirkungen auf den Prozess der Entwicklung und das Laufzeitverhalten. Recht verbreitet in Unternehmensanwendungen ist SQL (Structured Query Language), was die Möglichkeit bietet, komplexe Abfragen relativ effizient auszuführen.

## Navigation-API für C statt SQL

Das hohe Abstraktionsniveau von SQL macht die Abfragesprache zu einer »Black Box«: Der Entwickler hat kaum Möglichkeiten, die Leistungsfähigkeit und den Speicherverbrauch zu beeinflussen. Da SQL-Abfragen ein System erheblich verlangsamen können, ist dieses API für Systeme, die ein deterministisches Laufzeitverhalten erfordern, ungeeignet.



puffert das System häufig verwendete Daten in einem Cache, um schneller auf diese zugreifen zu können. Änderungen werden durch den Cache auf die Festplatte geschrieben. Ein neuerer Ansatz sind IMDS (In-Memory Datenbanksysteme), die keine Zugriffe auf die Festplatte ausführen und dafür alle Daten im Hauptspeicher ablegen. Die Daten schreibt das System nur dann auf die Festplatte, wenn die Anwendung dies verlangt.

## Scheibe oder Speicher?

In-Memory-Datenbanken bieten Persistenz, also die dauerhafte Speicherung von Daten über die Ausführungszeit von Programmen hinaus, mit Merkmalen wie Unterstützung von nichtflüchtigem Speicher (NVRAM), hohe Verfügbarkeit durch das Anlegen redundanter Datenbankkopien für ein Failover im Fehlerfall und Transaktionsprotokolle für das Wiederherstellen einer Datenbank.

Dual-Mode-Datenbanken kombinieren die Vorteile eines IMDS mit denen einer konventionellen Datenbank in einer Anwendung. Ein Beispiel dafür ist »eXtremeDB Fusion« von McObject. Sie spezifiziert einen Satz von Daten, die im Speicher und auf der Festplatte abgelegt werden. Das Ergebnis ist eine Datenbank, welche die Anforderungen an ein Embedded System hinsichtlich geringen Speicherbedarfs und intuitivem Programmier-Interface erfüllt. Einige Embedded-Datenbanken basieren auf dem Client/Server-Modell, das in Unternehmensanwendungen sehr verbreitet ist. Bei diesem Modell senden Client-Anwendungen Anforde-

Als Alternative zu SQL bieten viele Embedded-Datenbanken ein »Navigational API«, das eng an die Programmiersprachen wie C und C++ angepasst ist. Im Unterschied zu SQL arbeitet diese Art Programmierschnittstelle zu jeder Zeit mit nur einem Record. Ein wesentlicher Vorteil des »Navigational API« ist der Determinismus. Nach der Kompilierung ist bekannt, wie und wie schnell die Daten abgearbeitet werden. Von SQL bekanntes Parsing (die Anweisung zur Laufzeit interpretieren und den Zugriffspfad optimieren) ist hier überflüssig.

Datenbanken mit Navigational API sind »Berkeley DB« von SleepyCat (Oracle) und »RDM Embedded« (Birdstep Technology). Verschiedene Datenbanken bieten SQL und eine Navigational-Programmierschnittstelle. Beispiele dafür sind »eXtremeDB« (McObject), »RDM Server« (Birdstep Technology) und »c-treeSQL Server« (Faircom).

Die meisten Datenbanken – Enterprise und Embedded – sind Disk-basierend. Dabei

rungen an den Datenbank-Server. Auch wenn Client und Server auf dem gleichen Rechner sind, kommunizieren beide über einen Inter-Process-Kommunikationslayer. Jene Datenbanken, die ohne Client/Server-Modell auskommen, verwenden Funktionen in Java, C oder C++, die direkt in die Applikation kompiliert werden. Beispiele dafür sind Birdstep, McObject und Empress. Diese Architektur ist einfacher und erfordert auch wesentlich weniger Speicherplatz. Außerdem entfallen Server-Tasks für das Session- und Verbindungsmanagement sowie die Allokierung und De-Allokierung von Ressourcen für den Client.

Die Möglichkeit, mit sehr geringer oder keiner Administration auszukommen, ist ein großer Vorteil. Die Datenbank kann beispielsweise innerhalb eines Geräts der Unterhaltungselektronik kaum auf einen Administrator oder ein aufwändiges Management zurückgreifen. Das beschriebene »In-Process«-Datenbankmodell erfordert wesentlich weniger Management- und Entwicklungsaufwand als das Client/Server-Modell. Das Konzept des »Users« ist dabei unbekannt.

### **Anwendungen aus der Unterhaltungselektronik**

Aktuelle MP3-Player speichern sehr viele unterschiedliche Daten und ermöglichen einen schnellen Zugriff darauf. Neben den eigentlichen MP3-Dateien sind das auch Metadaten wie der Name des Künstlers, Name des Albums, Titel, Genre, Playlist und andere. Zu Beginn der Entwicklung erkannten die Spezialisten bei JVC, dass die Entwicklung einer proprietären Datenbanklösung für den »Alneo HD500« zu zeitaufwändig war.

Ein Ziel des Herstellers war es, den benötigten Speicher und die CPU-Ressourcen zu verringern; die Datenbank sollte schnellen Zugriff auf verwandte Records ermöglichen. Viele redundante Daten sind auch im Index der On-Disk-Datenbank abgelegt. Dieser beschreibt Beziehungen zwischen den Daten. Daten im Index können Zugriffe auf die Festplatte vermeiden. Mit Hilfe einer In-Memory-Datenbank konnte JVC die redundanten Indexdaten vermeiden und gleichzeitig den Speicherbedarf reduzieren.

Auch in der Automatisierungstechnik finden IMDS Einsatzgebiete, etwa bei Systemen für die Drucksteuerung bei der Förderung von Erdöl und Erdgas, wie sie beispielsweise Hydril herstellt. Dabei werden Daten von den Sensoren im Controller gespeichert und für die sichere und effiziente Steuerung der Förderanlagen verwendet. Eine Anforderung an die neue Datenbank war Hochverfügbarkeit (HV). Bei Datenbanken ist HV typischerweise über eine Replikation oder eine Konfiguration implementiert, bei der eine Datenbank (Master) die Änderungen an die anderen Datenbanken überträgt. Bei Ausfall der Master-Datenbank wird auf eine andere umgeschaltet. Eine passende Lösung war »eXtremeDB« von McObject mit HV-Unterstützung. Dabei erfolgt die Replikation der Datenbank innerhalb des Controllers, zusätzlich findet ein Abgleich der Datenbanken des gesamten Netzwerks mit jenen eines Standby-Netzwerks statt. (mc)

**Leon Houwen**  
ist Geschäftsführer von  
**Logic Technology**  
Telefon 089/99 21 64 34  
[www.logic.nl](http://www.logic.nl)

**ARS Software**  
Telefon 089/89 34 13 0  
[www.ars2000.com](http://www.ars2000.com)