Integration von TopLink in den Oracle 9iAS

Bernd Trops
Oracle

Schlüsselworte

O/R Mapping, Performance Entity Beans, Session Beans, EJB 2.0, Java Persistenz, J2EE, Servlets, JSP, JDO

Zusammenfassung

Der Vortrag erläutert die Funktionalitäten von Oracle 9iAS TopLink (in weiterem TopLink) und die Integration von TopLink in den Oracle 9iAS

Oracle 9iAS TopLink

TopLink ist der Weltmarktführer im objekt-relationalen Mapping, u.a. aufgrund seiner extrem flexiblen Einsetzbarkeit und den vielfältigen Performance-Tuning-Möglichkeiten. TopLink ist nun bereits seit 10 Jahren als Produkt erhältlich. Entstanden ist das Framework als Smalltalk Framework und wurde vor 5 Jahren nach Java migriert.
In diesem Vortrag wird die Funktionalität von TopLink und die Einbindung von TopLink in Oracle9iAS vorgestellt.



Abb. 1 Generelle Architektur

TopLink Architektur

TopLink ist ein Persistenz-Framework, das, wie aus der Abbildung ersichtlich ist, zwischen der Java-Applikation und einer relationalen Datenbank liegt.
Die drei Schichten sind unabhängig voneinander, d.h. ein Java-Objekt enthält keine Information über seine Persistenz: Diese Informationen stecken in der TopLink-Schicht; sie werden als Meta-Daten gehalten.
Diese Meta-Daten können über das mitgelieferte graphische Mapping-Tool, die Mapping Workbench, erstellt werden und als XML-Dateien oder als Java-Klasse zur Verfügung gestellt werden.
Durch diese Architektur gilt für TopLink:

- JEDE Java-Arckitektur
- JEDE Datenbank
- JEDER Applikations-Server

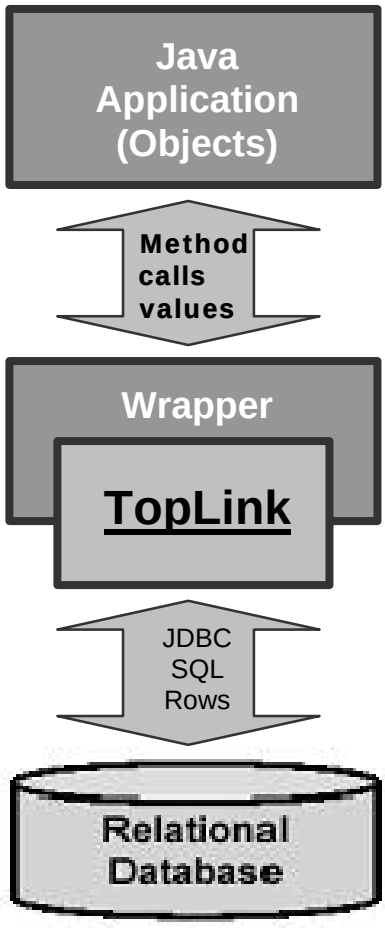


Abb. 2 TopLink Layer

Das schlanke Benutzer-Interface und die Architektur bieten viele Freiheitsgrade:

- Flexibilität der Abbildung (Mapping zwischen OO-Modell und Datenbank-Schema)
- Flexibilität der Datenbankwahl
- Viele Tuning-Möglichkeiten in dieser Schicht bzw. im direkten Zugriff

TopLink Features

Flexibilität
• Meta-Daten Architektur • 10 verschiedene Mappings • Reportqueries • Parametrisierbare SQLs • Alle gängigen Vererbungs-Patterns

Der Zugriff auf die Datenbank erfolgt über die TopLink-Session. Die Session wird mit den Meta-Daten initialisiert. Sie stellt eine einfache objekt-orientierte Schnittstelle für den Java-Entwickler zur Verfügung:

```
Session.readAllObjects(Person.class);
// Diesen Aufruf konvertiert TopLink in
// das folgende SQL-//Statement:
// SELECT * FROM PERSON.
```

Als Ergebnis kommt eine Collection von Personen-Objekten zurück.

```
Session.writeObject(meinPersonenObjekt);
```

Dieser Aufruf erzeugt je nach Kontext ein Update oder Insert – SQL-Statement auf das Objekt „meinPersonenObjekt“.

Diese beiden Zugriffsarten können in vielfältiger Weise erweitert werden.

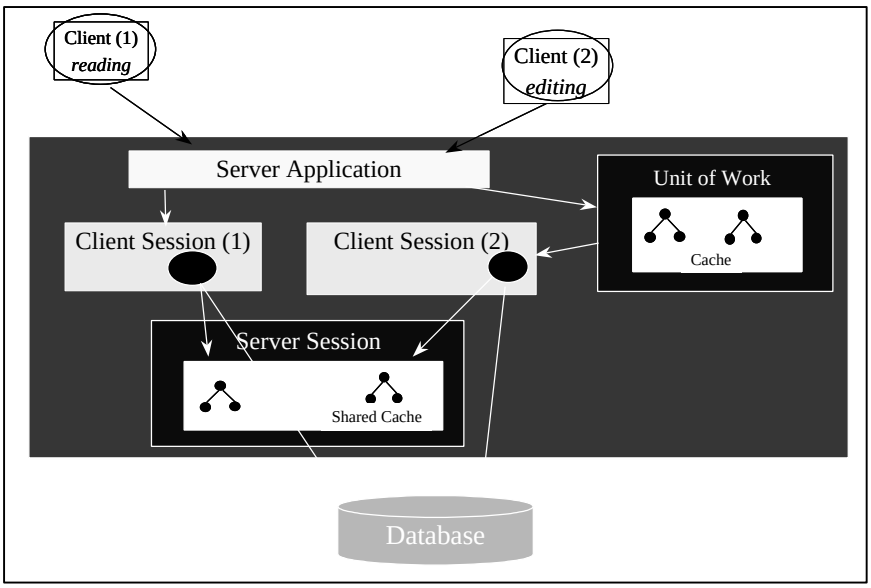


Abb. 3 TopLink Session

Mappings

Die Meta-Daten stellen die Informationen über die Mappings der TopLink Laufzeitumgebung zur Verfügung. Es stehen folgende Mapping-Typen zur Verfügung

Direct Mappings:	**Relationship Mappings:**
• Direct-to-Field • Type conversion • Object type • Serialized • Transformation • Objectrelational	• One-to-One • One-to-Many • Many-to-Many • Aggregate • Direct Collection

Mit Hilfe dieser Mappings kann jedes Objektmodel auf ein relational Datenbankschema abgebildet werden.

Queries

Des weiteren stellt das Framework sowohl eine objektorientierte Abfragesprache als auch die standardisierte EJBQL-Abfragesprache zur Verfügung.
Die Anwendung erhält als Resultat einer Query immer ein Java-Objekt oder einen Vektor von JavaObjekten zurück. D. h. die Anwendung muss sich nicht um das Extrahieren der Daten aus dem Resultset und das Aufbauen der Objekte kümmern.
Für Listenoperationen steht zusätzlich eine Reportquery zur Verfügung. Diese liefert der Anwendung eine Collection von Strukturen zurück. Wenn zum Beispiel von der Klasse Kunde nur die ID und der Name gewünscht wird, werden diese beiden Attribute für die ReportQuery definiert, die ReportQuery wird ausgeführt und als Ergebnis erhält man eine Collection die z. Bsp. als Input für eine Listbox zur Verfügung steht.

```
ExpressionBuilder builder = new
ExpressionBuilder();
ReportQuery query = new ReportQuery
(Kunde.class, builder);
query.addAttribute("nachName");
query.addAttribute("ID");
query.retrievePrimaryKeys();
Vector reportRows = (Vector)
session.executeQuery(query);
```

Performance Tuning

Tuningmöglichkeiten	
• 6 verschiedene Caching-Strategien • Minimale Updates • Joined Reading • Batch Reading; • Batch Writing • Caching von Statements	• Scrolling Cursors • Partielles Lesen von Objekten • Indirection bzw “Just in Time” Lesen • Suche im Cache • Abfrage-Optimierung • Sequencing • Unterstützung von StoredProcedures

In der obenstehenden Tabelle sind einige der Performance-Stellschrauben die das TopLink Framework bietet aufgelistet. All diese Möglichkeiten stehen zur Verfügung ohne das in die Businessklassen eingegriffen werden muss.
Diese Parameter wirken entweder direkt auf die Queries oder sind global im Mapping einstellbar.
Auch das Cachingverhalten von TopLink ist konfigurierbar. Der prinzipielle Ansatz des TopLink Frameworks ist über IdentityMaps die Eindeutigkeit von Objekten zu gewährleisten und somit sicherzustellen, das Objekte im Adressraum eindeutig referenzierbar sind.
Der zweite wichtige Punkt ist die Performance-Steigerung; die durch Caching erreicht werden kann. Befindet sich das Objekt bereits im Cache benötigt man keinen zweiten Datenbankzugriff und kann somit das Objekt der Anwendung sofort zur Verfügung stellen. Durch Vermeidung von Datenbankzugriffen kann die beste Performancesteigerung erreicht werden. Die verschiedenen Cachingstrategien definieren die Verweildauer der Objekte im Cache.
Auch „indirection“ ist ein wichtiges Feature von TopLink. Über „indirection“ wird festgelegt wie viel TopLink zu einer Zeit aus der Datenbank lesen soll. Über „indirection“

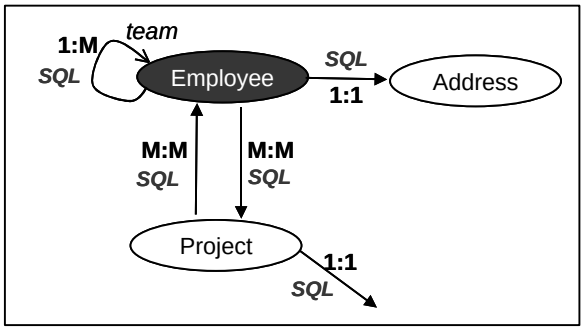


Abb. 4 „TopLink indirection“

kann an den Beziehungen zwischen Objekten definiert werden, ob ein in Beziehung stehendes Objekt mitgelesen wird oder nicht.
So besteht im Beispiel (s.Abb. 4) die Möglichkeit alle Beziehung von Employee, sprich die zu Adresse, Projekt oder auch zu Team als „indirection“ zu definieren. Wenn nun eine Query abgesetzt wird die die Attribute Name und Nachname des Angestellten

betrifft wird TopLink auch nur die Employee Tabelle abfragen. Erst wenn eine Methoden der Anwendung, z. Bsp. getAdress() das Objekt Address anfordert wird TopLink ein zusätzliches SQL-Statement erzeugen.

Zusätzliche Features

Zusätzliche Features	
• Verschiedene Optimistic-Locking Szenarien • Pessimistic Locking • Locking auf Cache-Ebene • Schnittstelle für nicht-relationale Datenquellen	• Automatische Feststellung von Änderungen • Cache-Synchronisierung • Transaktionsunterstützung und –integration • ApplikationsServer Unterstützung

Neben den bereits vorgestellten Features zur Flexibilität und zur Performance stehen in TopLink noch weitere Funktionalitäten zur Verfügung die in jeder Applikation genutzt werden.
Natürlich bietet TopLink eine Transaktionsverarbeitung an. Diese basiert auf einer objektorientierten Transaktionsverarbeitung. D. h. die Anwendung fordert von der TopLink Session ein Transaktionsobjekt an, an dieses registrieren sich jetzt die in der Anwendung befindlichen Objekte, es werden die Operationen auf den Objekten durchgeführt und zuletzt wird dem Transaktionsobjekt ein „commit“ gesendet.
Zu diesem Zeitpunkt ermittelt TopLink welche Veränderungen an den beteiligten Objekten durchgeführt wurden und erstellt, mit Hilfe der Meta-Daten, die entsprechenden SQL-Statements. Dabei werden beim Schreiben nur die Attribute betrachtet die auch geändert wurden. Nach dem die Änderungen ermittelt wurden eröffnet TopLink eine Datenbank Transaktion und führt die Statement auf der Datenbank aus.

TopLink arbeitet auch mit den entsprechenden Transaktionsmanagern der Applikations Server zusammen, darauf wird im nächsten Abschnitt noch eingegangen.
Im Zusammenhang mit Transaktionen steht natürlich auch das Locking. TopLink bietet alle Varianten von Locking an:

- Kein Locking,
- optimistisches Locking und
- pessimistisches Locking.

TopLink kann auch im Cluster betrieben werden. In diesem Fall gibt es einen TopLink Cache pro Applicationserver Instanz und diese Caches gleichen sich untereinander ab. Man spricht dann von Cachesynchronisation.

Einbindung in den Oralce9iAS

Wie anfangs beschrieben ist das TopLink Framework bereits vor der Zeit der J2EE Application-Server auf dem Markt gewesen. Aufgrund der Meta-Architektur von TopLink war es ein logischer Schritt auch die in J2EE beschriebenen EntityBeans zu unterstützen. Neben dieser Unterstützung können mit TopLink gemappte Objekte auch in SessionBeans genutzt werden. Dies wird in vielen Projekten genutzt da diese Variante, gerade bei Abfragen, die performantere Variante darstellt.
Im Oracle 9iAS steht TopLink in allen Varianten zur Verfügung. D. h. ob nun SessionBeans, bean managed persistence (BMP) für EntityBeans oder container managed persistence (CMP) eingesetzt werden soll, liegt einzig in der Entscheidung der Anwendungsdesigner. Es steht also die ganze Palette der möglichen Architekturen für die Anwendung bereit.
Des weiteren kann TopLink, wie bereits erwähnt, das Transaktionsmanagment des Oracle 9iAS nutzen. Die TopLink Transaktion hängt sich dann in die Oracle 9iAS Transaktion über JTS (Java Transaction Service) ein. Auch der JDBC ConnectionPool des Oracle 9iAS kann von TopLink genutzt werden.

Da sich TopLink an die in J2EE definierten Standards hält, kann TopLink auch mit WebLogic BEA, WebSphere IBM sowie mit allen J2EE zertifizierten Applikationsserver genutzt werden. BEA

WebLogic und IBM Websphere bieten lediglich container managed peristence (CMP) an. Als Anhang „A“ wurde eine tabellarische Gegenüberstellung angehängt.

Somit hat man durch TopLink gerade im Oracle 9iAS die Freiheit die Architekturvariante zu wählen, die für die Anwendung am Besten geeignet ist. TopLink unterstützt alle Varianten während die Applikationsserverhersteller ohne TopLink nur CMP anbieten können.

Kontaktadresse:

Bernd Trops
Principal Sales Consultant
Robert-Bosch-Str. 5
D-63303 Dreieich

Telefon: +49(0) 6103 397 231
Mobil: +49(0) 177 5944 231
E-Mail: bernd.trops@oracle.com
Internet: www.oracle.com; otn.oracle.com

http://otn.oracle.com/products/ias/toplink/content.html