

Externe OLAP Systeme unter Verwendung eines Oracle Datenbanksystems im Vergleich zu Oracle OLAP

Daniel Fritzler und Rüdiger Steffan
Fakultät für Wirtschaftswissenschaften
Hochschule Wismar

Schlüsselworte:

OLAP, Business Intelligence, Discoverer, Mondrian, JPivot, Palo, POET, CSS, eGECKO

Einleitung

Das Ziel von OLAP (online analytical processing) ist die Auswertung von großen Datenmengen in Echtzeit, während OLTP (online transactional processing) die Datenerfassung und Datenpflege verfolgt. Im Gegensatz zum Multidimensionalen OLAP (MOLAP) basierend auf gespeicherten Data Cubes, verfolgen verschiedene Softwaresysteme das Ziel, multidimensionale Strukturen außerhalb der Datenbank zu verwalten (Memory Based OLAP) und die Vorteile einer rein relationalen Speicherung zu nutzen. Darüber hinaus können auf diese Weise auch freie Datenbanksysteme wie Oracle XE eingesetzt werden, was insbesondere für kleine und mittlere Unternehmen aus Kostengründen interessant ist.

In diesem Beitrag sollen die Architektur und die Konfiguration verschiedener OLAP-Systeme mit einer Anbindung an ein Oracle Datenbanksystem zusammengefasst und mit den multidimensionalen Funktionalitäten der Oracle Enterprise Edition verglichen werden. Dazu zählen vor allem die Open Source-Produkte Mondrian/JPivot und Palo basierend auf Java. Durch die probeweise Migration von Beispieldaten sollen die Konfigurationen erläutert und Performance-Vergleiche durchgeführt werden. Darüber hinaus werden Problemfälle aus relationalen Systemen von Industriepartnern als Fallstudien vorgestellt und Auswahlkriterien für geeignete OLAP-Systeme diskutiert.

OLAP-Architekturen im Vergleich

Eine Übersicht über verschiedene OLAP-Systeme ist in [1] zu finden und in Tab. 1 zusammengefasst. Im Fall von Mondrian und Palo handelt es sich um Open Source Software (EGL und GPL).

OLAP Server	Unternehmen	MOLAP	ROLAP	HOLAP
Essbase	Oracle	X	X	X
icCube	MISConsulting SA	X		
Microsoft Analysis Services	Microsoft	X	X	X
MicroStrategy Intelligence Server	MicroStrategy	X	X	X
Mondrian OLAP Server	Pentaho		X	
Oracle Database OLAP Option	Oracle	X	X	X
Palo	Jedox	X		
SAP NetWeaver BW	SAP	X	X	X
SAS OLAP Server	SAS Institute	X		
Cognos TM1	IBM	X	X	

Tab. 1: Vergleich verschiedener OLAP-Server hinsichtlich dem Speicherungsprinzip (aus [1]).

Alle OLAP-Server in Tab. 1 unterstützen die multidimensionale Abfragesprache MDX, wobei es sich im Fall von Oracle um eine Zusatzsoftware für Excel, Cognos oder SAP handelt [2]. Mondrian basiert auf einem rein relationalen Speicherprinzip, stellt sich einer Anwendung jedoch als multidimensionales System dar, indem der OLAP-Cube im Arbeitsspeicher gebildet wird (siehe Abb. 1). Dieses speicherbasierte Prinzip ist beispielsweise auch bei IBM TM1 zu finden. Ein sehr ähnliches Ziel verfolgen Analyseprogramme wie Oracle Discoverer oder IBM Cognos (Framework Manager) über ein Repository (Metadaten) zur Definition von Fakten, Dimensionen und Hierarchien. Während Mondrian aufgrund des Zugriffs auf eine relationale Datenbank als ROLAP-System bezeichnet werden muss, speichert Palo die Daten in einem proprietären Format und kann als MOLAP-System gesehen werden. Im folgenden werden Mondrian und Palo anhand von Fallbeispielen näher betrachtet.

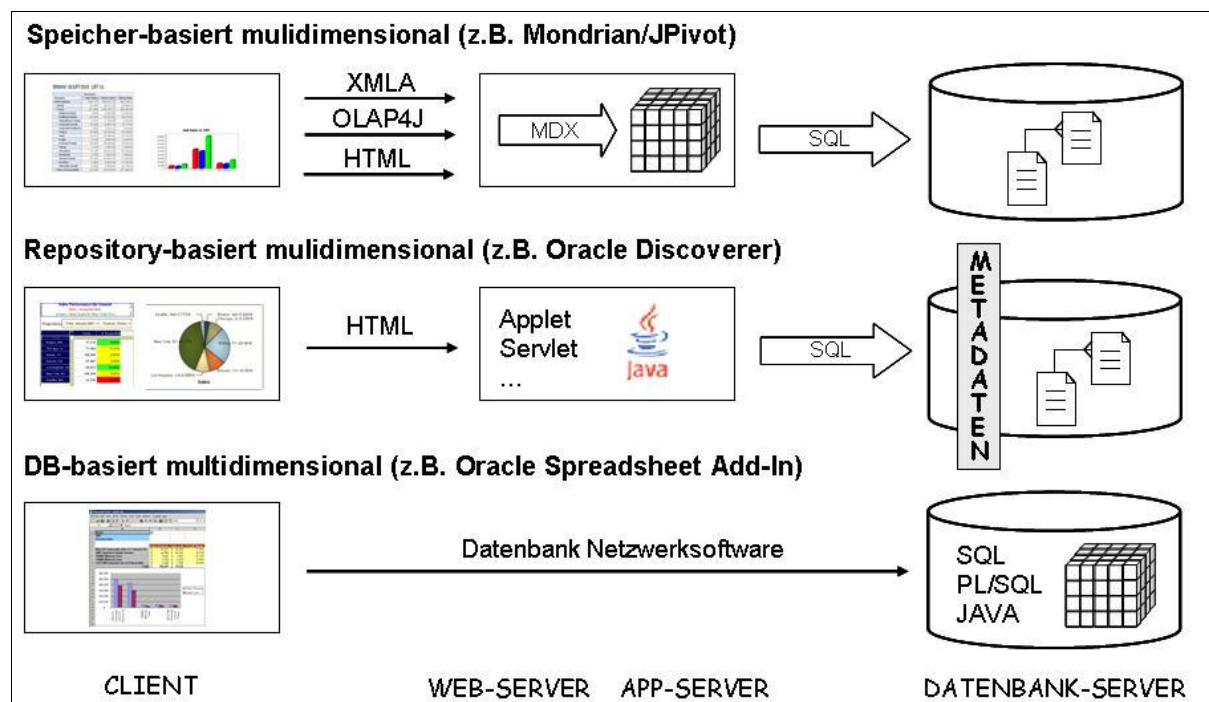


Abb. 1: Beispiele für Zugriffsprinzipien auf OLAP-Systeme aus Anwendungssicht.

Mondrian/JPivot und Beispielschemata

JPivot ist eine JSP-Programmbibliothek zur Abfrage und Darstellung von Inhalten des Mondrian OLAP-Servers. Dabei wird von beiden Seiten XMLA (XML for Analysis) unterstützt, das als Abfragesprache MDX verwendet [1]. Ein Beispiel für die Verwendung von JPivot ist in Abb. 3 im nachfolgenden Abschnitt zu sehen.

In einem weiteren Projekt, das in diesem Beitrag vorgestellt werden soll, werden die Einsatzmöglichkeiten von Mondrian und Palo zur OLAP-Erweiterung der betriebswirtschaftlichen Software eGECKO untersucht. eGECKO kann mit verschiedenen Datenbanksystemen eingesetzt werden, so dass eine herstellerunabhängige Lösung gefordert ist. Dennoch wäre auch der Einsatz von Oracle OLAP als OLAP-System für Oracle-Kunden denkbar. Um Unterschiede hinsichtlich Funktionalität und auch Performance zu untersuchen wurde zunächst das Beispielschema SH (sales history) als Mondrian Data Cube implementiert. Dies erfolgt in Form von XML-Dateien, was durch das Werkzeug Schema Workbench von Mondrian unterstützt wird (siehe Abb. 2).

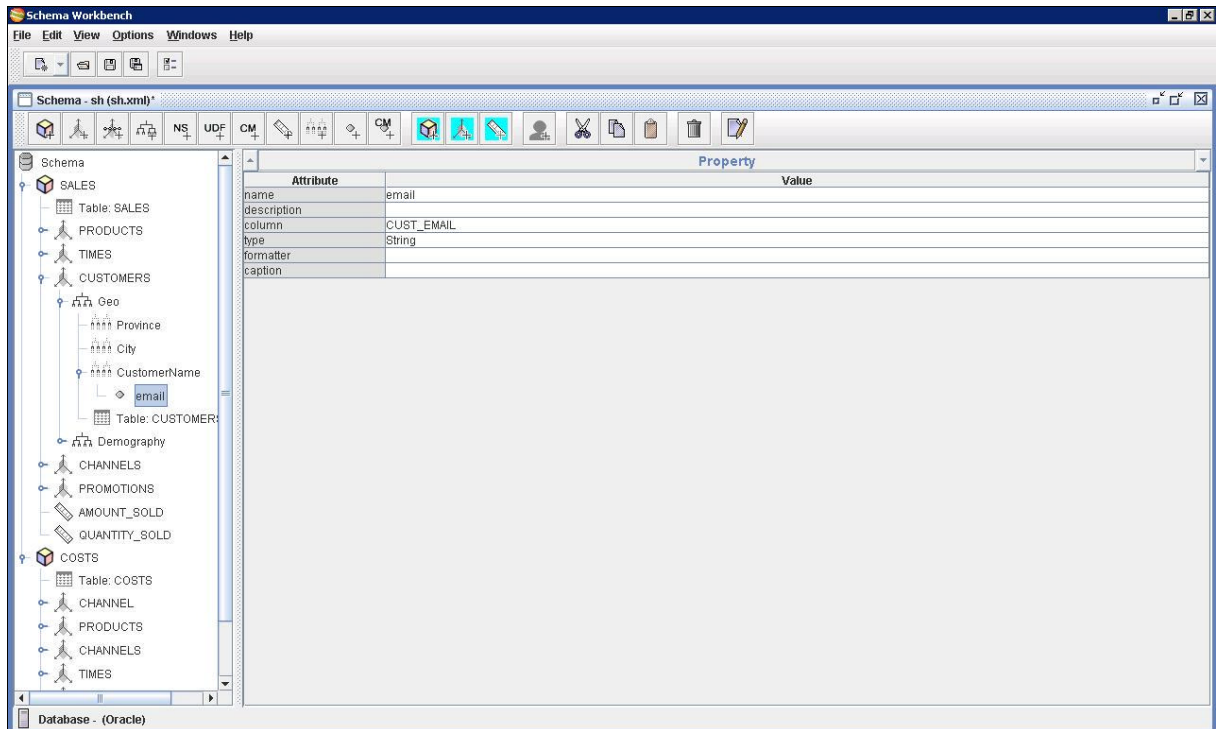


Abb. 2: Mondrian Schema Workbench zur Definition des Data Cubes für das Oracle Beispielschema SH.

Fallbeispiel: Katalogmanagement

Die POET AG ist einer der Marktführer im Katalogmanagement für den B2B-Onlinehandel [4]. Im Rahmen einer Masterarbeit an der Hochschule Wismar wurde der Prototyp eines Moduls für die Katalogplattform POET X-Solutions unter Verwendung von Mondrian/JPIVOT basierend auf einem Oracle Datenbanksystem entwickelt [5]. Ein Beispielbericht ist in Abb. 3 zu sehen.

POET X-SOLUTIONS®

► HOME | ► CATALOG | ► DISTRIBUTION | ► CONFIGURE | ► ADMINISTRATION | ► HELP

Catalog > Financial Dashboard > FD Query Result

Search: Search

Compare Quotations 1
Compare quotations for a specified request. The view is group by classification and products. Measures for order volume effect in relation to old suppliers.

Classification	Product	5 most ordered products from 2009/08/05 V.1			rfq Supplier 3 - quotation - b V.1			rfq Supplier 2 - quotation - b V.2		
		Order volume	Absolute volume effect	Relative volume effect	Order volume	Absolute volume effect	Relative volume effect	Order volume	Absolute volume effect	Relative volume effect
All by name	All Products	€ 1,635,530.05	0	0%	€ 1,284,550.13	€ -350,979.92	-21.5%	€ 1,419,744.24	€ -215,785.81	-13.2%
	sup1_EXT_MD1lc_b	€ 508,750.00	0	0%	€ 325,600.00	€ -183,150.00	-36.0%	€ 350,020.00	€ -158,730.00	-31.2%
	sup1_BPP600_b	€ 473,843.70	0	0%	€ 382,963.10	€ -90,880.60	-19.2%	€ 407,505.58	€ -66,338.12	-14.0%
	sup1_BIW1400_b	€ 350,482.00	0	0%	€ 290,900.06	€ -59,581.94	-17.0%	€ 301,414.52	€ -49,067.48	-14.0%
	sup1_EXT_DBS100_b	€ 186,929.60	0	0%	€ 177,278.13	€ -9,651.47	-5.2%	€ 221,431.64	€ 34,502.04	18.5%
	sup1_EXT_DBS25_b	€ 115,524.75	0	0%	€ 107,808.84	€ -7,715.91	-6.7%	€ 139,372.50	€ 23,847.75	20.6%

Abb. 3: Prototyp eines Moduls für POET X-Solutions unter Verwendung von Mondrian/JPIVOT mit Oracle [5].

Fallbeispiel: Finanzbuchhaltung und Controlling

eGECKO ist eine Software der Firma CSS GmbH zur Unterstützung der Unternehmensbereiche Rechnungswesen, Personalwirtschaft, Controlling und CRM. Basierend auf der Java-Technologie ist eGECKO betriebssystem- und datenbankunabhängig. Auf Datenbankebene wird deshalb auf herstellerspezifische Methoden wie etwa Constraints oder Materialized Views möglichst verzichtet. Nicht zuletzt aus diesen Gründen kommt es bei der Auswertung aggregierter Daten zu Performanceeinbußen. Statt für jedes Datenbankmanagementsystem eine Performancestrategie zu implementieren wird derzeit untersucht, in welchem Ausmaß sich durch die Integration eines OLAP-Servers in das eGECKO-System die erwünschte Performancesteigerung realisieren lässt [6]. Die Integration soll so nahtlos verlaufen, dass es für Anwender nicht sichtbar ist, ob die Daten vom OLTP- oder vom OLAP-System stammen. Die Wahl des OLAP-Servers fiel auf Mondrian, da dieser ebenfalls betriebssystem- und datenbankunabhängig ist und im Gegensatz zu Palo auch in der freien Lizenz die Anbindung mittels XMLA unterstützt. Das Prinzip der Architektur ist in Abb. 4 dargestellt.

Da in eGECKO Buchungsdaten bereits in Form eines Snowflake-Schemas vorliegen, kann auf den Schritt der Denormalisierung verzichtet werden. Mit Hilfe des graphischen Mondrian-Werkzeugs Schema Workbench wurde ein Data Cube in einer XML-Datei definiert. Mittels OLAP4J werden MDX-Abfragen aus der Java-Umgebung heraus an den OLAP-Server geschickt. Bestehende SQL-Abfragen zu aggregierten Daten werden derzeit probeweise in MDX nachgestellt.

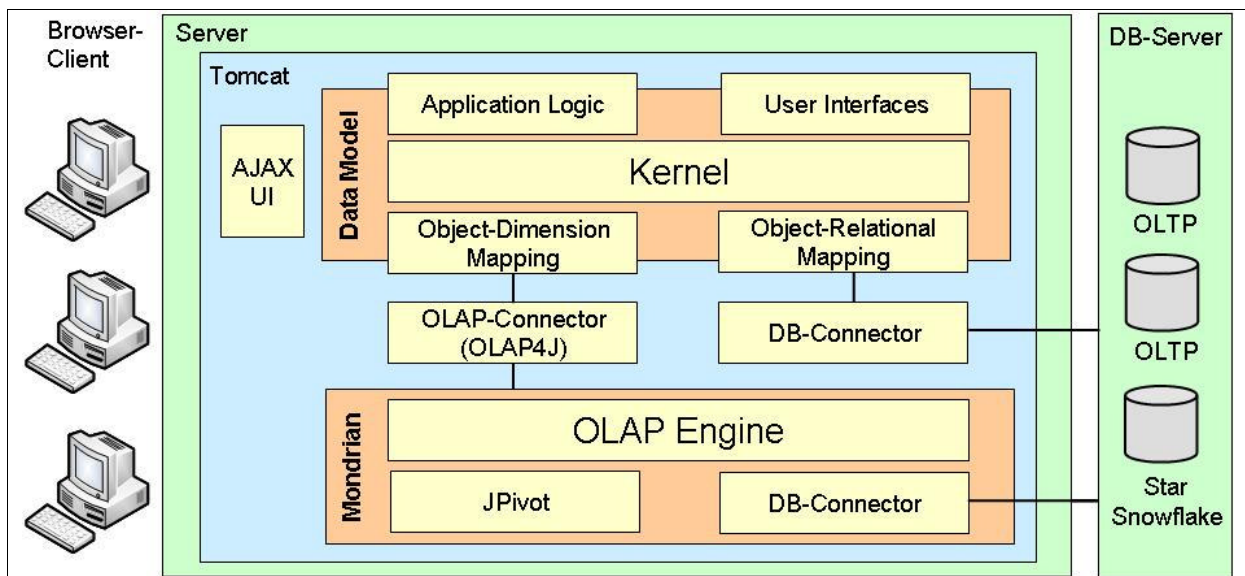


Abb. 4: Architektur von eGECKO und Konzept zur Integration von Mondrian/JPivot [6]. Im Fall eines Rich-Clients befindet sich die Software mit der Swing UI anstelle von AJAX UI auf der Client-Seite und greift ebenso mittels OLAP4J über das Netzwerk auf Mondrian zu.

Um dem Benutzer die Flexibilität eines intuitiven OLAP-Systems bieten zu können, muss eine Methode gefunden werden, wie aus Benutzerinteraktionen MDX-Abfragen generiert werden. Dazu werden die Möglichkeiten der Open Source-Produkte JPivot und Saiku untersucht. Ebenso wird eine auf eGECKO zugeschnittene Klasse benötigt, welche eine Kreuztabelle repräsentiert (Object-Dimension-Mapping analog zum bisherigen Object-Relational Mapping). Darüber hinaus werden die Möglichkeiten der Performanceoptimierung von Mondrian mit sogenannten Aggregated Tables, Cache-Management und Materialized Views untersucht. Für größere Kunden käme es jedoch auch in Frage, eine Lösung unter Verwendung der Oracle OLAP Option bereitzustellen. Aktuelle Ergebnisse und Erfahrungen werden in diesem Beitrag vorgestellt.

Zusammenfassung

Es konnte gezeigt werden, dass der OLAP-Markt eine Fülle von Produkten mit unterschiedlichen Architekturen und Strategien bietet. Auf Grund der vielfältigen Methoden zur Performanceoptimierung ist es nicht möglich, eine grundsätzliche Aussage zur besten Strategie zu machen. Vielmehr sollten Zugriffsmöglichkeiten und die Integrierbarkeit in das bestehende System höher gewichtet werden. Wird etwa Datenbankunabhängigkeit vom Gesamtsystem erwartet, so sollte diese auch vom OLAP-System verlangt werden können.

Quellenangaben

[1] Internetquellen: Beispiele für OLAP-Systeme

http://en.wikipedia.org/wiki/Comparison_of_OLAP_Servers

<http://www.oracle.com/technetwork/database/options/olap/>

<http://mondrian.pentaho.com/> <http://jpivot.sourceforge.net/>

<http://www.jedox.com>

[2] Oracle Press Release: Oracle and Simba Technologies Introduce MDX Provider for Oracle OLAP

<http://www.oracle.com/us/corporate/press/036550>

<http://www.simba.com/MDX-for-Oracle-OLAP-documentation.htm>

[3] Rüdiger Steffan und Julia Neumann-Szyszka: Fallstudie Einkaufscontrolling: BW trifft WI – ein Erfahrungsbericht, Datenbank-Spektrum, 9. Jahrgang, Heft 29, 2009.

[4] Internetquellen: Fallbeispiele für Mondrian/JPivot

<http://www.poet.de>

<http://www.css.de>

[5] René Marzahl: Modulentwicklung für eine E-Procurement-Software zur Realisierung von Angebotsanfragen mit einem bestelldaten-basiertem Angebotsvergleich, Masterarbeit Hochschule Wismar, 2010.

[6] Daniel Fritzler: Erweiterung des Systems eGECKO mit OLAP-Funktionalität, Praktikumsbericht, CSS GmbH Fulda, Hochschule Wismar, 2011.

Kontaktadresse:

Rüdiger Steffan, Prof. Dr.-Ing.

Lehrgebiet Datenbank- und Datenkommunikationssysteme

Fakultät für Wirtschaftswissenschaften, Hochschule Wismar

University of Technology Business and Design

Phillip-Müller-Straße 14

23966 Wismar

Telefon: +49(0) 3841-753606

Fax: +49(0) 3841-7539606

E-Mail: ruediger.steffan@hs-wismar.de

Internet: www.wi.hs-wismar.de