

# Caching In-Memory Column Store oder im BI Server

Andreas Buckenhofer  
Daimler TSS GmbH  
Ulm

## Schlüsselworte

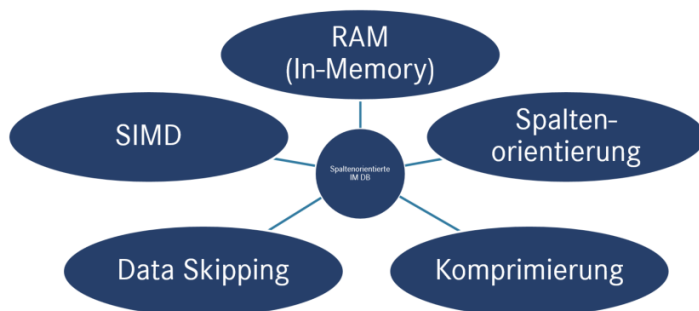
Cube, In-Memory, IMDB, Cognos Dynamic Cubes, SAP Hana, IBM, DB2 BLU, MS SQL Server Columnstore Index, DWH Architektur

## Spaltenorientierte In-Memory Konzepte

In-memory Konzepte sind nicht neu. Das Caching von Daten im Buffer Cache wurde schon immer durchgeführt, um teure IO-Zugriffe zu vermeiden.

Spaltenorientierte Konzepte sind ebenfalls nicht neu. Datenbanken wie Sybase IQ oder Vertica entstanden in den 2000er Jahren. Erst durch SAP HANA entstand in den letzten Jahren ein großer Hype um spaltenorientierte In-Memory Datenbanken.

Als weitere wichtige Konzepte sind die Komprimierung von Daten, Data Skipping sowie Parallelisierung / SIMD zu nennen.



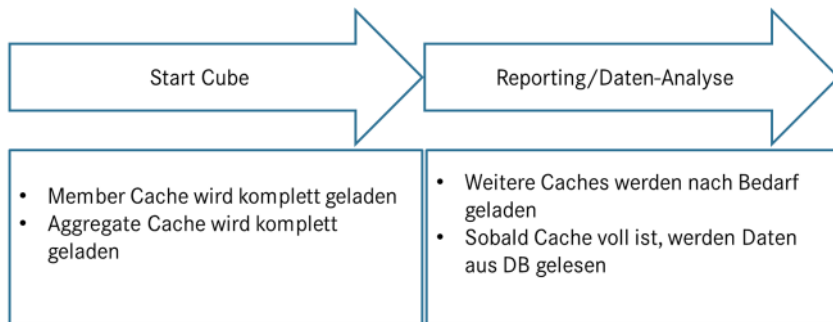
Alle großen Hersteller wie IBM DB2, MS SQL Server, Oracle oder SAP Hana verfügen über eine spaltenorientierte In-Memory Komponente. Die oben aufgeführten Konzepte setzen all diese Hersteller um, jedoch unterschieden sich die Hersteller teilweise sehr deutlich. In der nachfolgenden Tabelle sind einige der Unterschiede aufgeführt. Bereits beim Anlegen (create table vs create index) unterscheiden sich die Hersteller – es gibt keinen einheitlichen Standard.

	SQL Server	IBM DB2 BLU	Oracle IMDB	SAP HANA
<b>Erstintführung</b>	07/2014	06/2013	04/2014	10/2010
<b>In-Memory</b>	OLTP und OLAP	OLAP	OLAP	OLTP und OLAP

<b>Befehl zur Erzeugung der spaltenorientierten Tabelle</b>	Create index ...	Create table ...	Eigenschaft einer Tabelle	Create table ...
<b>Datenkomprimierung</b>	Ja	Ja	Ja	Ja
<b>Data Skipping</b>	Ja	Ja	Ja	Ja
<b>Datenhaltung</b>	<ul style="list-style-type: none"> <li>• Zeilen (In-Memory Organized Tables)</li> <li>• Spalten (Column store Index)</li> </ul>	<ul style="list-style-type: none"> <li>• Zeilen und Spalten (Column-organized Tables oder Schattentabellen)</li> </ul>	<ul style="list-style-type: none"> <li>• Zeilen und Spalten simultan (Buffer Cache für Zeilen)</li> </ul>	<ul style="list-style-type: none"> <li>• Zeilen und Spalten (Spaltenpräferenz)</li> </ul>
<b>In-Memory Analytics / R</b>	Ja	Nein	Ja	Ja
<b>Persistenz Columnstore</b>	Ja	Ja	Nein	Ja
<b>Aggregate Pushdown</b>	Ja (ab 2016)		Ja	
<b>Limitierung durch RAM</b>	Nein	Nein	Nein	Nein (nicht mehr)
<b>SIMD</b>	Ja	Ja	Ja	Ja

Im BI-Umfeld wurden in der Vergangenheit häufig OLAP Cubes entwickelt, um interaktive Analysen durchzuführen. Im Falle von ROLAP handelt es sich um die Speicherung der Daten als Star-schema in der Datenbank. Beim MOLAP werden dagegen multidimensionale Strukturen aufgebaut und die Kennzahlen vorberechnet für eine performante Analyse. Im Falle von Cognos Dynamic Cubes werden Daten in verschiedenen Caches im Hauptspeicher gehalten:

- Member Cache für Dimensionen und Hierarchien
- Aggregate Cache für Vorberechnungen
- Data Cache für einzelne, elementare Zellen
- Expression Cache für Ergebnisse von Auswertungen
- Result Cache für fertige Ergebnisse

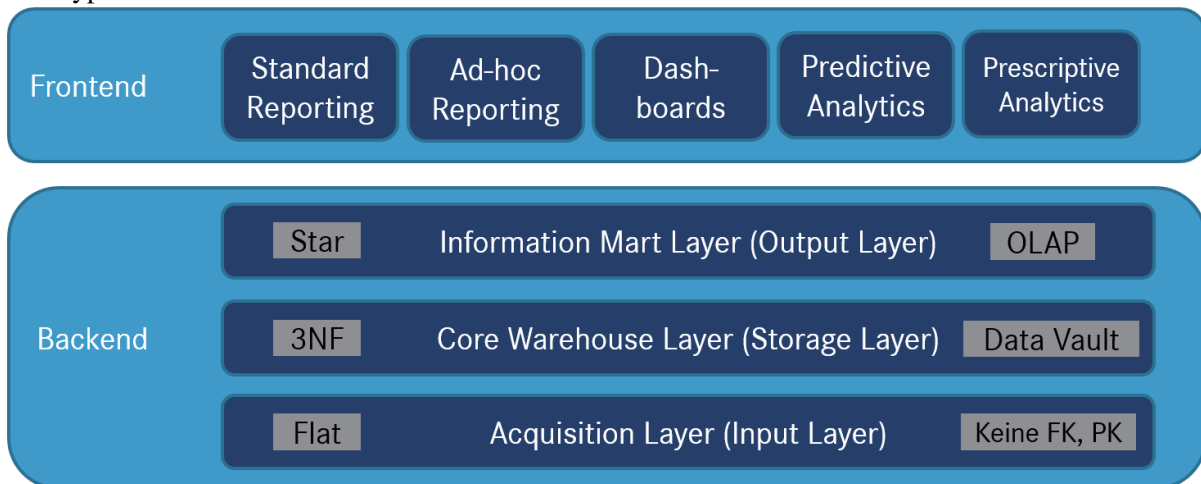


Cognos InMemory Cubes werden zu Beginn in den Cache geladen (Member + Aggregate Cache). Danach steht ein Cube zur Verfügung. Die Beladung der einzelnen Caches erfolgt aus Daten der relationalen DB. Wird ein Cube neu aufgebaut (z.B. nach einem Servercrash oder aufgrund eines kompletten Datenrefreshes), so werden die Daten erneut aus der DB geladen.

Die Performanz von In-Memory Datenbanken machen den Einsatz von OLAP Cubes immer weniger notwendig. In-memory Datenbanken sind wesentlich flexibler und vermeiden den bei OLAP Cubes häufig verwendeten nächtlichen Batch zur Beladung. Außerdem ist zusätzlicher Aufwand zur Realisierung der Cubes notwendig sowie deren Abfrage z.B. mittels MDX.

Es ist zu erwarten, dass OLAP Cubes weniger Anwendung finden werden. OLAP Cubes eignen sich noch vereinzelt für größere Datenmengen mit vorberechneten Werten oder für komplexere/rechenintensive KPIs. Ein Vorteil kann die Lizenzierung sein – ist Cognos BI lizenziert, so sind die Cubes automatisch auch in der Lizenz enthalten. Ad-hoc Reporting oder Datenvirtualisierung (Data Marts aus Views) sind dagegen mit OLAP Cubes nicht zu erreichen. Auch die Flexibilität (Bsp.: beschränkt auf die Auswertung numerischer Kennzahlen) ist deutlich geringer im Vergleich zu Datenbanktabellen und SQL.

Eine typische DWH-Architektur umfasst Backend und Frontend.



OLAP Cubes werden dem Information / Analytics Mart Layer zugeordnet. Spaltenorientierte In-Memory Datenbanken können in allen drei Layern vorkommen: am Üblichsten ist deren Einsatz im Information / Analytics Mart Layer bzw im Core Warehouse Layer.

**“In 10 years there will be no row stores in the data warehouse world.”** erwartet Stonebraker, der an der Entwicklung von Datenbanken wie Ingres, PostgreSQL, Vertica, VoltDB, SciDB beteiligt war.

Das Potential spaltenorientierter Datenbanken ist groß, jedoch lassen Stabilität, Ladeperformanz und Funktionsumfang bei den meisten Herstellern aktuell noch zu wünschen übrig. Es ist notwendig eine Anwendung zu testen, ob die spaltenorientierte In-Memory DB einen Vorteil bringt. Bei einem ungünstigen Design oder Workloads, die das spaltenorientierte Format nicht unterstützen (z.B. Zugriff auf genau einen Datensatz, Lesen einer ganzen Zeile) ist kein Vorteil zu erwarten.

Auch bei der Komprimierung muss genau geprüft werden, ob bestimmte Tabellen größer werden.

Tabellen mit wenigen Spalten, die dann auch noch unique sind, ist mit größerem Platzbedarf zu rechnen – je nach Komprimierungsart, z.B. Komprimierung einer Data Vault Hub-Tabelle mit Hash-Keys

Größe in GB	Verfahren
9 GB	Buffer Cache, Basic compression
9.7 GB (größer!)	IMDB, FOR QUERY LOW compression
8.4 GB	IMDB, FOR QUERY HIGH compression

**Kontaktadresse:**

Andreas Buckenhofer

Daimler TSS GmbH

Business Unit Analytics

Wilhelm-Runge-Straße 11

89081 Ulm, Germany

Telefon: +49-(0)731/505-6345

E-Mail [Andreas.Buckenhofer@daimler.com](mailto:Andreas.Buckenhofer@daimler.com)

Internet: <http://www.daimler-tss.com>