

# Cache vs. ‚InMemory‘

## Begriffe, Modellierung, reale Probleme

Frank Effenberger

Robotron Datenbank-Software GmbH

Dresden

### Schlüsselworte

InMemory, Cache, Business Intelligence, Data Warehouse, Modellierung, Performance

### Einleitung

Während die zeilenorientierte Speicherung von Datenbankobjekten im Hauptspeicher (Cache) bereits seit längerer Zeit bei Oracle verfügbar ist, bietet die Oracle 12c Datenbank mit der InMemory Option zusätzlich die Möglichkeit der spaltenorientierten Datenhaltung.

Dies verspricht vor allem im Bereich der Business Intelligence und des Data Warehouse eine Performancesteigerung. Die Möglichkeit, gleichzeitig zeilen- und spaltenorientiert innerhalb einer Datenbank zu arbeiten, bietet dabei klare Vorteile. So werden z. B. nur wenige SQL-Befehle und kein aufwändiges Migrationsprojekt dafür benötigt. Manche Versprechen des Marketings werden jedoch nur bedingt eingehalten und müssen kritisch diskutiert, sowie die Formulierungen entwirrt werden.

Der Vortrag wird zum einen die Begrifflichkeiten verständlich darstellen und erklären, dass InMemory nur situationsbedingt wirkliche Performancesteigerungen bringt. Des Weiteren wird aufgezeigt, dass die eigentliche Frage nicht der prinzipielle Einsatz des Hauptspeichers ist, sondern ob und wann zeilen- oder spaltenorientiert gearbeitet werden sollte und diese Varianten verglichen werden müssen.

Aus diesem Grund wird erklärt, in welchen Fällen der **Cache** (zeilenorientiert) eine sinnvolle Alternative zu ‚**InMemory**‘ (spaltenorientiert) im Hauptspeicher ist und die Anwendungsfälle der spaltenorientierten Speicherung aufgezeigt. Weiterhin wird anhand der Oracle Business Intelligence Enterprise Edition 11g gezeigt, dass Oracle sein eigenes Potential in diesem Themengebiet noch nicht vollständig nutzt.

### Testumgebungen

Die Datengrundlage des Vortrags entstand aus zwei Testsystemen. **Testsystem 1 (klein)** hatte eine Oracle Release 12.1.0.2-Datenbank und **2 GB RAM**, der nur für InMemory adressiert wurde. Dieses kleine Testsystem wurde für die Erstellung von Theorien bzw. erste Vermutungen eingerichtet und anschließend durch das größere Testsystem 2 validiert.

**Testsystem 2 (groß)** hatte eine Oracle Release 12.1.0.2-Datenbank und **64 GB RAM**, der nur für InMemory adressiert wurde. Die ersten Vermutungen des Testsystem 1 wurden durch einen zweiten, unabhängigen Tester bestätigt.

## Vorgehen

Um Tabellen in der Oracle Datenbank 12c spaltenorientiert in den Hauptspeicher zu laden, muss neben dem für die SGA zugewiesenen Speicher auch der Parameter *INMEMORY* gesetzt werden und für die aktivierte Session (oder global als System) der Parameter *INMEMORY\_QUERY* auf *ENABLE* gesetzt werden. Danach können Tabellen über das Stichwort *INMEMORY* (*NO INMEMORY* zum deaktivieren) in den Hauptspeicher geladen werden. Zur Verdeutlich folgendes Beispiel als SQL-Befehle:

```
SELECT * FROM V$SGA; -- Überprüfung, ob Speicher reserviert ist
SHOW PARAMETER INMEMORY_SIZE; -- Überprüfung, ob Parameter gesetzt ist
ALTER SESSION set inmemory_query = enable; -- Aktivierung der InMemory-Abfragen
ALTER SYSTEM FLUSH SHARED_POOL; -- Bereinigung der Speicherbereiche
ALTER SYSTEM FLUSH BUFFER_CACHE; -- Bereinigung der Speicherbereiche
ALTER TABLE X INMEMORY; -- Vorbereitung auf InMemory
SELECT * from X; -- Laden der Tabelle in den Hauptspeicher
```

Wichtig ist, dass eine Tabelle nur dann in den Hauptspeicher geladen wird, wenn die erste Abfrage darauf ausgeführt wurde. Wenn ein sofortiges Laden der Tabelle in den Hauptspeicher gewünscht wird, ist dies über den Parameter *PRIORITY CRITICAL* zu steuern. Weitere Informationen bezüglich der aktuellen InMemory Parameter sind über folgende Abfrage zu erhalten:

```
SELECT TABLE_NAME, INMEMORY, INMEMORY_COMPRESSION, INMEMORY_DISTRIBUTE,
INMEMORY_PRIORITY, INMEMORY_DUPLICATE
FROM USER_TABLES
ORDER BY TABLE_NAME;
```

Neben der spaltenorientierten Speicherung ist es auch möglich, dass die Tabellen zeilenorientiert im Cache gehalten werden. Dies wird standardmäßig gemacht und der zugewiesene Cache muss nur als Parameter hoch genug eingestellt sein. In der Regel finden daher Vergleiche der Testwerte zwischen gecachten Abfragen in zeilenorientierter und spaltenorientierter Variante statt. Zusätzlich kann über den Parameter *CACHE* (*NO CACHE* zum Deaktivieren) gesteuert werden, dass eine Tabelle als ‚most recently used‘ gilt und somit nicht aus dem Cache entfernt wird. Dies ist über folgenden Befehl möglich:

```
ALTER TABLE X CACHE;
```

## Ergebnisse

Die Aussage, dass InMemory prinzipiell schneller ist, basiert auf dem Vergleich einer spaltenorientierten Hauptspeichertechnologie mit normalen Festplatten. Der dadurch zu sehende Performancegewinn ist irreführend, denn auch wenn man den bereits vorhandenen Cache mit einer Festplatte vergleicht, wird man einen deutlichen Performancegewinn feststellen. Es ist daher wichtig,

dass der zeilenorientierte Cache mit dem spaltenorientierten InMemory von Oracle verglichen wird, um richtige Aussagen treffen zu können. Die Vorteile der Hauptspeichertechnologien bilden den Wegfall von physischem I/O und bedeuten zeitgleich eine Verschiebung des Bottlenecks auf die CPU. Dies ist anhand der physical reads der folgenden Abbildung erkenntlich (siehe **Abbildung 1**).

### Statistiken

```
-----  
335678 consistent gets (NO INMEMORY)  
34321  physical reads (NO INMEMORY)  
...  
252    consistent gets (INMEMORY)  
0      physical reads (INMEMORY)  
...  
4430634 rows processed
```

Abbildung 1: Abfragestatistiken

Von daher sollte beim Einsatz von Hauptspeichertechnologien auch dementsprechend die Anzahl der Kerne und damit ggf. Lizenzierungskosten überprüft werden. Anhand von zwei Extrembeispielen wird verdeutlicht, dass der Zeitfaktor (*Dauer gecachte Abfrage / Dauer InMemory-Abfrage*) stark variieren kann. Dabei gilt, dass ein höherer Zeitfaktor bedeutet, dass die InMemory-Abfrage schneller als die gecachte Abfrage war (siehe **Abbildung 2**).

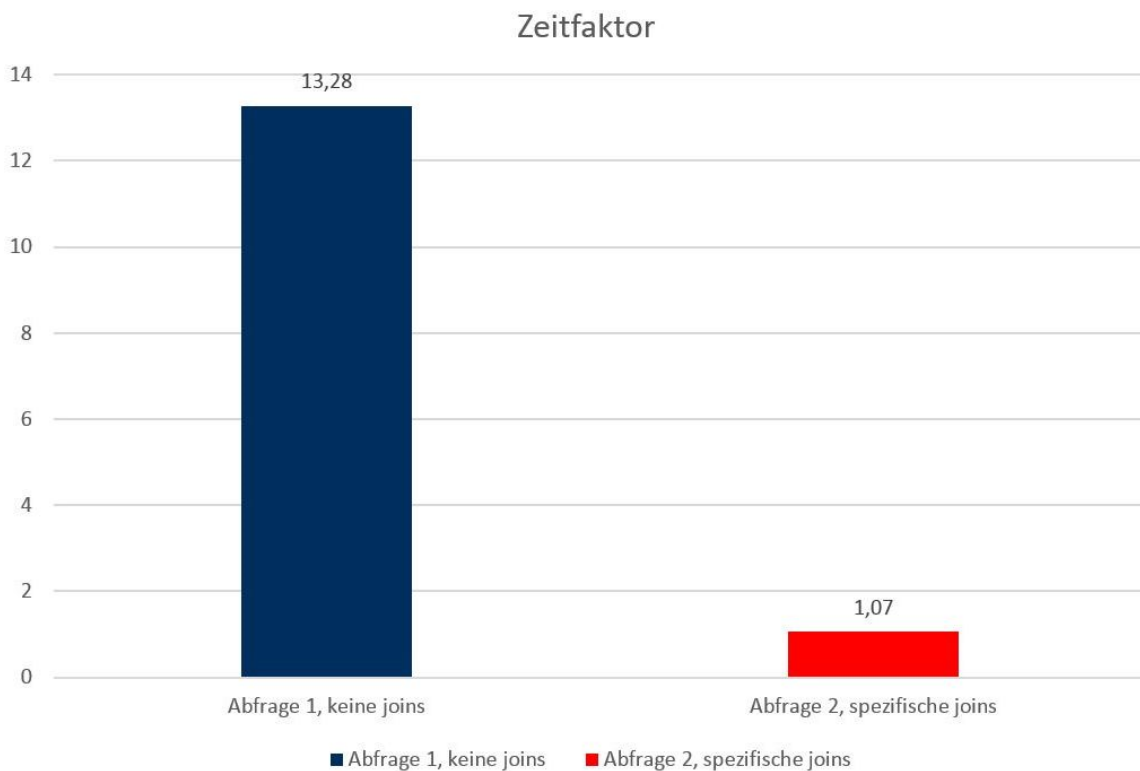


Abbildung 2: Abfrageperformance Cache vs. InMemory

Abfrage 1 bestand aus der Bildung der Summe dreier Spalten einer Faktentabelle und Abfrage 2 bestand aus der Bildung der Summe dreier Spalten derselben Tabelle, zusätzlich verknüpft mit zwei Dimensionstabellen. Dabei wurde die Where-Bedingung so erstellt, dass explizit eine Zeile jeder Dimensionstabellen gewählt und nur wenige Datensätze der Faktentabelle abgefragt wurden.

Durch die Tests wurde festgestellt, dass InMemory nur in bestimmten Situationen Performancegewinne bringt. Es müssen SQL- und Codereviews durchgeführt werden, um Optimierungspotentiale abzuschätzen und eine Kosten- Nutzenrechnung aufgrund des hohen Preises der InMemory-Option erstellt werden. Der Cache muss situationsbedingt als Alternative betrachtet werden.

Die Oracle BI-Suite Enterprise Edition Plus (11g) ist noch nicht in der Lage, InMemory bei komplexen Abfragen auszunutzen. Das Optimierungspotential liegt dabei im BI-Server, der aufgrund seiner z. T. sehr komplexen Abfragen ein spaltenorientiertes System dazu zwingt, zeilenorientiert zu arbeiten. Damit werden im Moment die Vorteile von InMemory negiert. Es ist abzuwarten, ob und wie in der Oracle BI-Suite Enterprise Edition Plus 12c InMemory Abfragen durch den BI-Server auch bei komplexeren Abfragen unterstützt werden, ohne dass ein großer Entwicklungsaufwand nötig ist.

**Kontaktadresse:**

M.Sc. Frank Effenberger  
Robotron Datenbank-Software GmbH  
Stuttgarter Straße 29  
01189 Dresden  
Telefon: +49 351 25859-2512  
Fax: +49 351 25859-3699  
E-Mail [frank.effenberger@robotron.de](mailto:frank.effenberger@robotron.de)  
Internet: [www.robotron.de](http://www.robotron.de)