

# **Oracle 12cR1 In-Memory Real World POC and what's coming up with 12cR2**

**Christian Pfundtner  
DB Masters GmbH  
Stammersdorfer Str. 463  
2201 Gerasdorf**

## **Schlüsselworte**

Oracle 12c, In-Memory, POC, Real World, Column Store, Oracle 12cR2, Data Guard, Standby, Active Data Guard, Kundenprojekt

## **Einleitung**

In diesem Vortrag werden einerseits unsere Erfahrungen mit Oracle 12c In-Memory bei POC von Kunden behandelt, aber auch ein Ausblick auf die geplanten Erweiterungen der In-Memory Technologie in der kommenden Version 12.2 dargestellt.

## **Teil 1 – Oracle 12c In-Memory Architektur Überblick**

Für den Fall, dass noch jemand die Architektur von Oracle 12c In-Memory nicht kennt, behandeln wir die wichtigsten Fakten als Einleitung.

Damit man die In-Memory Technologie sinnvoll einsetzen kann, braucht man ausreichend (zusätzlichen) Hauptspeicher, der ca. 20-50% der Größe der Tabellen, die man mit In-Memory nutzen möchte, sein muss. Im Fall von virtualisierten Systemen braucht man CPUs mit Built-In Virtualisierungstechnologie, die auch eingeschaltet sein muss (BIOS). Im Fall von Intel CPUs wird VT-x und VT-d benötigt.

Die In-Memory Technologie von Oracle setzt auf einen, zusätzlich zum vorhandenen Buffer Cache eingerichteten In-Memory Column Store. Dessen Größe muss man definieren und dann die Instanz neu starten. Danach kann man definieren welche Tabellen (Partitionen) in den In-Memory Column Store geladen werden. Das Laden wird von einer Reihe von Hintergrundprozessen erledigt.

Im Gegensatz zum Buffer Cache, wo die Datenblöcke 1:1 mit dem Abbild auf der Disk vorhanden sind und die Daten in ZEILEN (ROWS) gespeichert sind, werden die Daten im In-Memory Column Store in sogenannten IMCUs (In-Memory Column Units) abgelegt. Dabei werden die Daten SPALTENWEISE angeordnet. In jeder IMCU sind für einen Teil (Block Range) der Tabelle alle Spalten entsprechend „komprimiert“ angelegt. Jeder IMCU kennt für jede Spalte den kleinsten sowie größten vorkommenden Wert. Zusätzlich sorgt ein Verzeichnis mit HASH Werten für eine Liste aller vorkommenden Werte für kurze Zugriffswege.

DML erfolgt ganz normal gegen die Daten im Buffer Cache, es werden jedoch zusätzlich alle modifizierten Datensätze im In-Memory Column Store auf STALE gesetzt sowie die Transaktionsinformation in einen speziellen Teil des In-Memory Column Stores geschrieben. Hintergrundprozesse sorgen dafür, dass diese Datenänderungen in den IMCUs nachgezogen werden. Trifft eine Query auf einen STALE Datensatz im In-Memory Column Store, wird automatisch auf den entsprechenden Datensatz im Buffer Cache zugegriffen → das Ergebnis ist somit immer korrekt (aber vielleicht nicht ganz so schnell wie erwartet).

In-Memory eignet sich optimal für analytische Queries und Data Warehouse Queries (FACT und DIMENSION Tabellen), vor allem dann wenn der Zugriff mit „Full Table Scan“, „Full Index Scan“, „Fast Full Index Scan“ oder „Index Range Scan“ erfolgt.

## **Teil 2 – In-Memory Real World POC von Kunden**

Disclaimer: Die Kunden wollen aus verschiedenen Gründen nicht genannt werden, lediglich die Ergebnisse und Erkenntnisse sind in diesem Vortrag enthalten.

### **SAP HR – POC**

Der erste POC mit In-Memory sollte ein SAP HR System beschleunigen, leider war das absolut nicht möglich, da SAP HR eine Applikation ist, die aus den folgenden Gründen nicht von der In-Memory Technologie profitieren kann:

- Die meisten Abfragen greifen nur auf wenige Datensätze zu (oft über Primary/Unique Key)
- Die Queries dauern meist nur ms oder maximal 1-2 Sekunden
- Aber: Die Antwortzeit für den Benutzer liegt im Bereich von vielen Sekunden bis hin zu Minuten, weil das SAP System selbst so lange für die Verarbeitung der Daten benötigt.

### **Zusammenfassung zu SAP HR – POC**

Wenn eine Applikation gerade einmal 5% der Gesamtverarbeitung in der Datenbank zu tun hat, kann keine Technologie das Problem in der Datenbank lösen. Selbst wenn die Abfragen SOFORT die Ergebnisse liefern, würden beim Benutzer nur maximal 5% Performanceverbesserung ankommen → das merkt man nicht. SAP HR ist somit kein Kandidat für die In-Memory Technologie.

### **SAP BW/BI – POC**

In diesem POC haben wir die TOP (ca. 10) Queries aus der Produktion genommen und diese gegen die In-Memory Technologie von Oracle laufen lassen. Zusätzlich haben wir einige Beispielqueries selbst erzeugt, um unsere Erwartungen an die In-Memory Technologie zu verifizieren.

Die Datenbank wurde zuerst gecloned und dann auf Oracle 12.1.0.2 hochgezogen. Dabei haben wir auch die Auswirkungen von „großen“ Buffer Caches im Vergleich zu In-Memory durchgeführt. Folgende Ergebnisse wurden ausgewertet:

- Aktuelle Laufzeit der Queries im Produktionssystem (mit 20GB Buffer Cache) und unter Produktionslast
- Die Laufzeit der Queries auf dem Testsystem mit 20GB Buffer Cache
- Die Laufzeit der Queries auf dem Testsystem mit 256GB Buffer Cache
- Die Laufzeit der Queries auf dem Testsystem mit 20GB Buffer Cache und 80GB in-Memory Column Store

Alle Queries wurden mindestens 3 mal hintereinander durchgeführt (und die Tabellen mit CACHE in den Buffer Cache geladen) um sicher zu stellen, dass der Buffer Cache immer „warm“ war. Gewertet wurde das beste Ergebnis. Auf dem Produktionssystem wurde die mittlere Laufzeit für das Statement verwendet.

Da die Daten im Column Store komprimiert angelegt werden, haben wir die verschiedenen Einstellungen für die Komprimierung im Vorfeld ausprobiert. Der POC ist dann mit COMPRESS FOR QUERY LOW durchgeführt worden. Abhängig von den Tabelleninhalten waren diese mehr oder weniger gut komprimierbar. Die Komprimierung für QUERY sorgte für Faktoren zwischen 2 und 5.

## Ergebnisse / Erkenntnisse auf Grund einer Beispielquery

```
select count(*) from
  (select /*+ FULL(s) */ *
    from SAPEBI."/BIC/EEG_BCO_12" s );
```

Die Tabelle belegte ca. 11GB auf der Disk und ca. 3.7GB im Column Store. Da die Query alle Spalten selektiert und keine where clause hat, ist diese dazu geeignet den reinen Memorydurchsatz zu vergleichen. Die Laufzeit betrug mit Buffer Cache ca. 3 Sekunden und mit In-Memory ca 660ms. Das wäre ein Faktor 5. Da der In-Memory Column Store durch Komprimierung nur ca. 1/3 des Platzes belegt, war ein Faktor 3 zu erwarten. Der restliche Unterschied geht auf den Overhead des „ROW Processing“ beim Buffer Cache zurück.

Wie könnte man diese Query schneller machen?

Nun, zuerst einmal nur jene Spalten selektieren, die man wirklich benötigt. WHERE Clauses mit Einschränkungen, die nur einen Teil der Daten angreifen sowie Aggregationen und Gruppierungen helfen ebenfalls, die Abfrage schneller zu machen – und genau das passiert bei analytischen Queries! Zusätzlich kann man den Zugriff auf den In-Memory Column Store durch Parallelisierung deutlich skalieren, wobei das parallel Degree von der CPU Technologie und der Memoryanbindung abhängt. Bei unseren Tests mit 2 Sockel Servern und CPUs mit 4 Memory Bussen pro CPU war eine Parallelisierung mit 4 oder 8 meist am „effizientesten“ (höchste Skalierungsfaktoren). Am schnellsten war meist eine Parallelisierung mit 16, wo hingegen eine Parallelisierung mit  $\geq 32$  praktisch immer schon merklich langsamer als mit 8 Prozessen gelaufen ist.

Wenn man dies bei der Query entsprechend umsetzt, zeigt In-Memory seinen Vorteile:

```
select /*+ FULL(s) */ sum(KEY_EG_BCO_12P), sum(KEY_EG_BCO_127)
  from SAPEBI."/BIC/EEG_BCO_12" s;
```

Bei dieser Variante der Query lief diese gegen das Buffer Cache mit ca. 7 Sekunden und bei in Memory mit 70ms – somit um den Faktor 100 schneller. Wenn man jetzt noch sinnvolle WHERE Clauses verwendet, liegt In-Memory noch weiter vor dem traditionellen Zugriff.

Ergebnisse aus dem POC (in Sekunden)

- Original Laufzeit auf dem Produktionssystem mit 11.2.0.x und 20GB Buffer Cache 2.746
- Oracle 12c mit 20GB Buffer Cache 1.500
- Oracle 12c mit 256GB Buffer Cache 380
- Oracle 12c mit 20GB Buffer Cache und 80GB In-Memory 100

Die Werte für 12c sind gerundet auf 10 Sekunden, da die Laufzeitschwankungen im Bereich von 5-10% lagen, war eine genauere Auswertung nicht sinnvoll.

Zusammenfassung SAP BW/BI POC

Der Einsatz der In-Memory Technologie in diesem Bereich von SAP ist wirklich zu empfehlen. In der Zwischenzeit hat Oracle mit dem SAP BW-EML Benchmark gezeigt, dass Oracle mit der In-Memory Technologie schneller ist als SAP HANA.

## **In-Memory POC für eine DWH Applikation**

Vorgangsweise für den POC:

- Clonen der Datenbank auf einen anderen Server (mit den identen HW Specs) und Upgrade auf 12.1.0.2
- Die TOP 30 Queries aus dem Produktions-DWH heranziehen
- Droppen aller Indexes, die nicht zu PK, UK und FK gehören.
- Alle von den 30 Queries angegriffenen Tabellen in den In-Memory Column Store legen
- Test des täglichen ETLs für einen Tag

Ergebnisse der Queries:

- Die meisten – aber leider nicht alle – wurden ohne weiteres Tuning zumindest um den Faktor 2-3 schneller, einige sogar über Faktor 100.
- Jene Queries, die nicht schneller wurden, wurden insofern optimiert, dass nur jene Spalten selektiert wurden, die für das Ergebnis relevant waren. Dadurch wurden diese zumindest nicht langsamer als bisher.
- Tests mit verschiedenen Parallelsierungsgraden haben gezeigt, dass ein parallel Degree von 4 oder 8 meist am besten war. Parallel 16 hat oft im Vergleich zu parallel 8 nur noch wenige % Performancevorteil bedeutet, aber die CPU entsprechend mehr belastet.

Ergebnis des ETL

- Der ETL Prozess wurde nicht fertig (in dem Zeitraum, den er auf den Produktionssystem benötigt). Der Grund waren fehlende Indexes für die ETL Verarbeitung (wir haben ja viele Indexes gedroppt).
- Nach dem Anlegen der benötigten Indexes war der ETL Prozess in ca. 25% der Zeit fertig, die er auf dem Produktionssystem benötigt hatte. Der Grund dafür war ganz einfach, dass viel weniger Indexes (darunter auf Bitmap Indexes) zu pflegen waren.

Zusammenfassung des POC für eine DWH Anwendung

- In-Memory brachte für (fast) alle Queries eine teilweise dramatischen Performanceboost.
- ETL wurde durch das Eliminieren von nicht mehr benötigten Indexes deutlich schneller
- Der Tuningaufwand für die Queries und ETL ist vernachlässigbar im Vergleich zum traditionellen Ansatz.

### Teil 3 – was kommt mit Oracle 12cR2 und In-Memory

Auf der Open World wurde ein Ausblick auf die kommende Oracle 12cR2 gegeben. Alle folgenden Informationen sind lediglich Ankündigungen, es kann jedoch sein, dass Oracle diese bis zur Release von 12cR2 noch ändert oder gar streicht.

#### Faster In-Memory Joins

Für Joins müssen die Daten aktuell „decompressed“ werden. In der nächsten Version kann man sogenannte Join Groups definieren, die dann ohne diese Dekomprimierung direkt joinen können. Das soll ca. einen Faktor von 3-5 bringen.

#### In-Memory Expressions (virtual columns)

Es wird möglich sein, Berechnungen (Beispiel: Brutto = Netto + Steuern) in virtuellen Columns anzulegen und auf diese so zuzugreifen als ob diese schon existieren. Dieses Konzept gibt es auch schon in Oracle 11g, allerdings nicht im Zusammenhang mit In-Memory. Auch hier soll ein Faktor 2-5 an Performance zu holen sein.

#### JSON und In-Memory

Ab 12.2 wird JSON direkt in einem In-Memory optimierten Format verarbeitet. Je nach Zugriff auf JSON Tabellen und Werte soll dies 10-60 mal schneller sein als aktuell.

#### In-Memory Column Store auch für Active Data Guard Instanzen

Ab 12.2 soll es möglich sein für Datenbank „Services“ entsprechende Tabellen in den In-Memory Column Store zu legen. Sprich: da die Produktionsinstanz einen anderen Servicennamen wie die Data Guard Instanz hat, kann man für Produktion und Data Guard verschiedene Objekte in den In-Memory Column Store legen. Aktuell wird der In-Memory Columnstore nur auf Primary Instanzen unterstützt.

#### Automatic Data Optimisation für In-Memory

Ab 12.2 wird die Heat Map Technologie auch dazu genutzt, Tabellen (Partitionen) bei Bedarf in den In-Memory Column Store zu legen oder aber auch zu löschen. Dadurch entfällt viel vom manuellen Aufwand für die Entscheidung welche Objekte zu welchem Zeitpunkt im In-Memory Column Store liegen sollen.

#### In-Memory Advisor Integration

Es gib aktuell via OTN Download einen In-Memory Advisor, der soll in 12.2 sowohl in die Datenbank als auch in den OEM integriert sein.

#### SQL in Silicon mit SPARC M7

Durch eine spezielle „Database Acceleration Engine“ pro Core im M7 werden einige Funktionen direkt auf der CPU ausgeführt. Dazu gehören:

- Der native Support von „Basic Database Query Primitives“ – beispielsweise das finden von „California“ in einer Spalte im In-Memory Column store
- Native OZIP dekomprimieren von In-Memory CAPACITY komprimierten Daten on the fly (mit >120GB/sec) – dadurch belegt man weniger Platz im In-Memory Column Store ohne mehr CPU zu benötigen.

- Support für In-Memory Format im Flash Cache von Exadata Storage Nodes – damit kann man In-Memory Queries in die Exadata Storage Cells auslagern.

Abschließende Worte:

Oracle In-Memory Technologie ist schon heute eine wirklich interessante Möglichkeit um analytisches Queries und DWH Anfragen zu beschleunigen. Der Ausblick auf 12cR2 zeigt, dass diese Technologie noch Einiges zu bieten hat.

Wir von DB Masters stehen Ihnen auch gerne jederzeit hilfreich zur Seite, wenn Sie Themen rund um die Oracle Datenbank haben.

**Kontaktadresse:**

Christian Pfundtner

DB Masters GmbH  
Stammersdorfer Str. 463  
2201 Gerasdorf  
Österreich

Telefon: +43 699 15037884

E-Mail [cp@dbmasters.at](mailto:cp@dbmasters.at)

Internet: [www.dbmasters.at](http://www.dbmasters.at)