Optimiertes Laden in die F-Fakten-Tabelle des SAP BW

Jörn Bartels
Oracle
München

Schlüsselworte

SAP BW Index unusable.

Einleitung

Mit Oracle Database 11g Release 2 kann das Laden der F-Fakten Tabelle in einem SAP BW System optimiert werden. Bisher musste gegen existierende Indizes geladen werden, oder sie wurden vorher gelöscht und nach dem Laden wieder neu aufgebaut. Mit dieser Optimierung kann der neu Aufbau der Indizes um den Faktor 10 verbessert werden, sowie auf die Tabelle weiterhin optimal zugegriffen werden. Es werden die Vorteile des Index Löschen sowie des Index Erhalten kombiniert.

Standard Laden der F-Fakten Tabelle – Laden gegen Indizes

Eine normale F-Fakten Tabelle hat auf jeder Partition mehrere lokale Indizes. Wenn ein Request in eine Partition geladen wird, dann müssen alle Index Partitionen auf diesen Daten-Partitionen permanent mitgeschrieben werden.

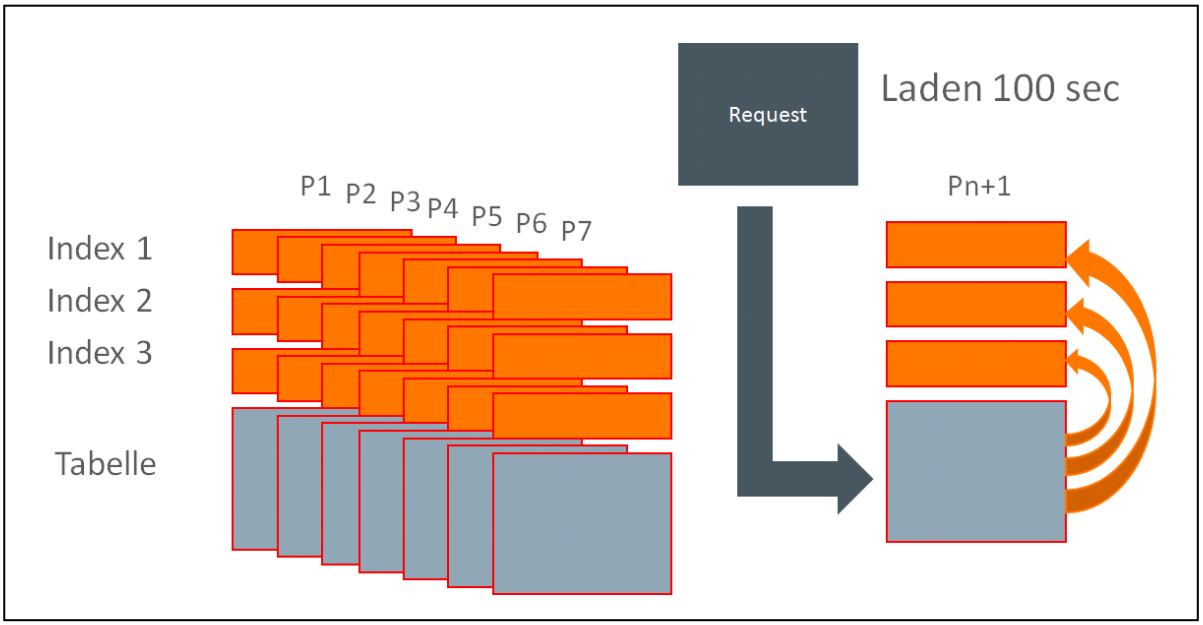

Abb. 1: Laden gegen vorhandene Indizes

Der Aufwand um diese Indizes zu pflegen ist weitaus höher als die Daten in die Tabelle zu schreiben.

In unserem Beispiel dauert das Laden der Tabelle ohne Indizes 6 Sekunden und mit Indizes sind es 100 Sekunden. Das ist ein Faktor von mehr als 16, um den die Indizes das Laden verlangsamen.

Löschen und Wiederaufbau der Indices

In einer ersten Optimierung versucht man sich den Vorteil des schnellen Ladens ohne Indizes zu Nutze zu machen. Bevor der Ladevorgang angestossen wird, werden alle Indizes auf der Tabelle gelöscht, und nachher wieder angelegt.

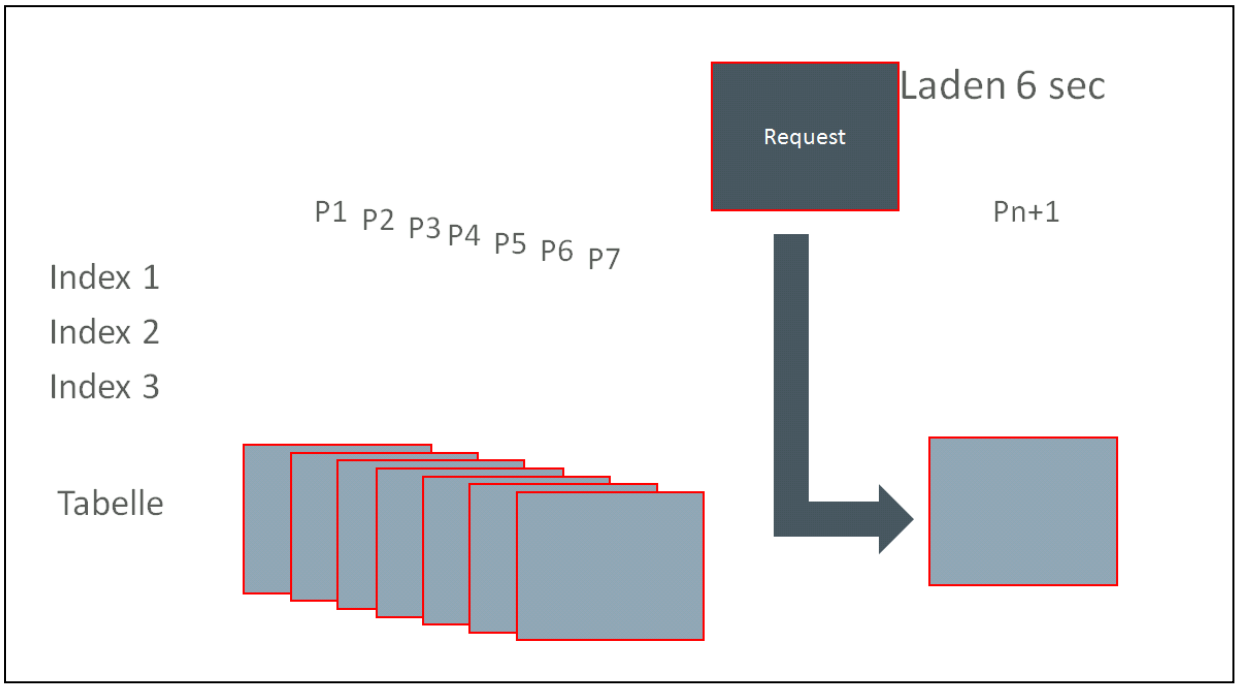

Abb.2 : Laden ohne Indizes

Jetzt muss jedoch auch die Zeit berücksichtigt werden, die der Wiederaufbau der Indizes benötigt. Es müssen alle Indizes auf allen Partitionen neu angelegt werden, und die dauert in unserem Beispiel 200 Sekunden.

Dieses Vorgehen ist daher für das Laden eines einzelnen Requests nicht sinnvoll, da insgesamt eine Zeit von 206 Sekunden gebraucht wird. Wenn man jedoch mehrere Request zusammen lädt, kann eine Verbesserung der Gesamtlaufzeitt erreicht werden.

Bei einer Zusammenfassung von 10 Requests in einem Ladelauf benötigt das Laden gegen die vorhandenen Indizes 10 * 100 Sek. = 1.000 Sek., während das Laden ohne Index mit anschliessendem Index Neuaufbau 10 * 6 Sek. + 200 Sek. = 206 Sek. benötigt. Das ist eine Verbesserung um den Faktor 5.

Man darf jedoch gravierende Nachteile dieser Methode nicht verschweigen:

Die Tabelle ist während des Ladens nicht für Zugriffe über Index verfügbar, sondern alle Zugriffe müssen über einen Full Table Scan erfolgen. Dies ist bei größeren Tabellen natürlich illusorisch, da die Zugriffszeiten viel zu lang werden.

Der Erfolg dieser Methode hängt direkt von der Größe der Tabelle ab. Wenn es viele große Partitionen gibt, dann dauert der Wiederaufbau der Indizes entsprechend lange.

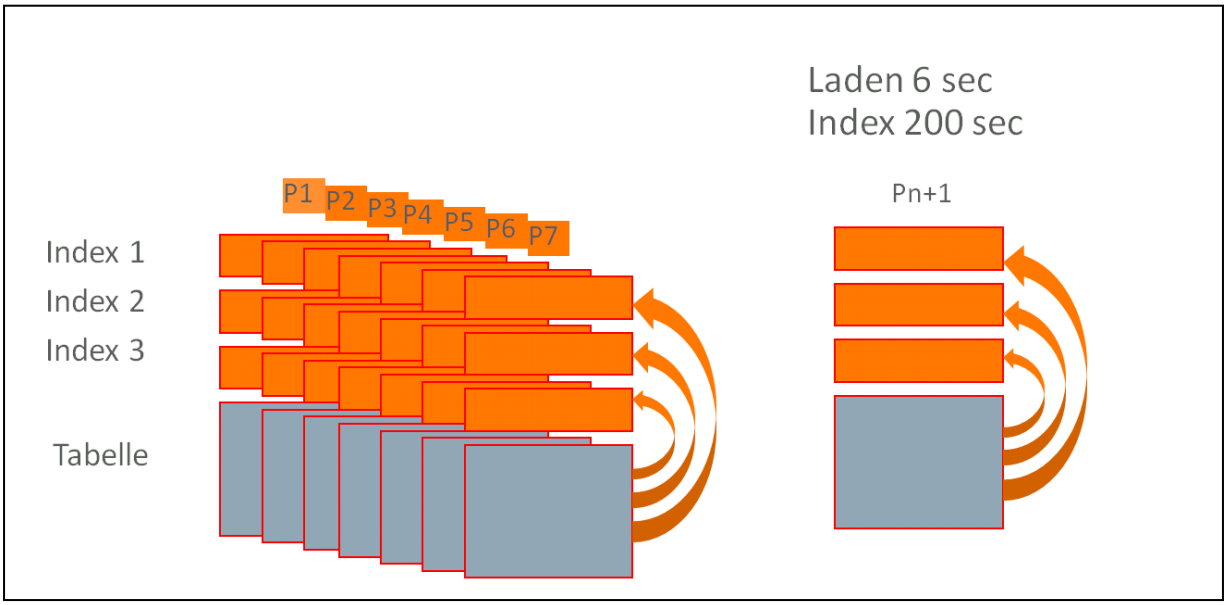

Abb. 3: Wiederaufbau aller Index Partitionen

Optimierung des Ladens durch Unusable Index Partionen

Mit unusable Index Partitionen kann der Nachteil der langen Index Aufbau Zeiten vermieden werden. Es wird jeweils nur die Partition der einzelnen Indizes auf unusable gesetzt, die zu der gerade zu ladenden Daten Partition gehört.

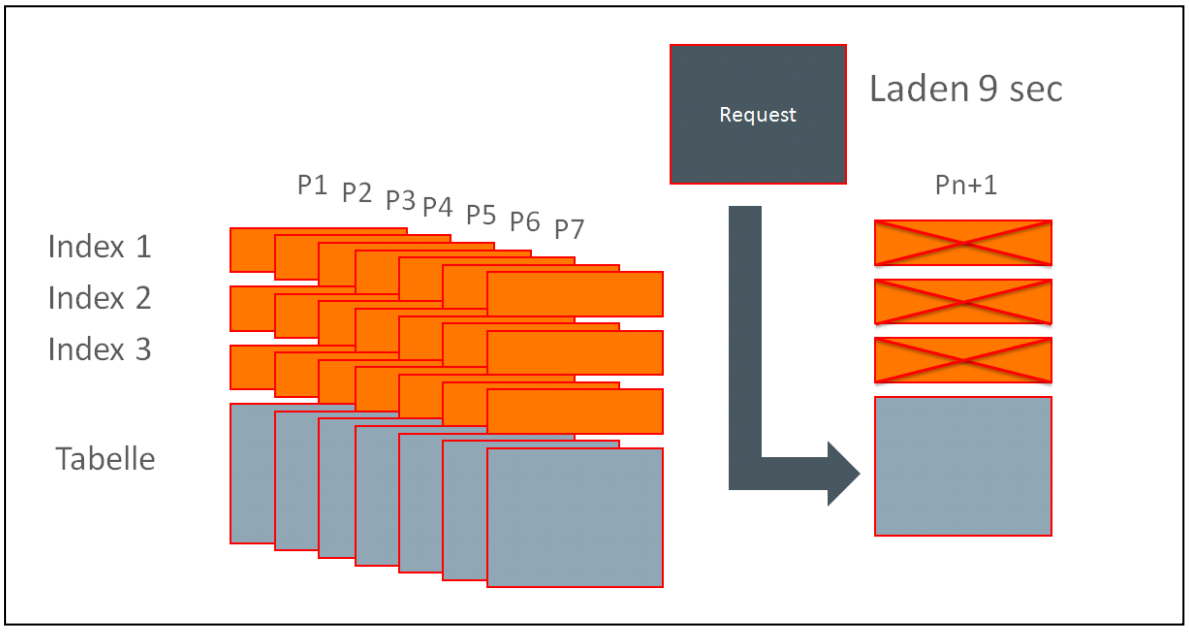

Abb. 4: Laden mit unusable Index Partition

Sobald eine Partition eines Index auf unusable steht wird diese nicht mehr mitgepflegt, und das Laden der Datenpartition ist dadurch fast genau so schnell wie das Laden ohne Indizes..

Diese Methode vermeidet den Nachteil der langen und unkalkulierbaren Index Aufbauten, da immer nur die lokalen Indizes für einzelne Partitionen neu aufgebaut werden.

Der Befehl um eine Index Partition auf unusable zu setzen sieht wie folgt aus:

```
ALTER INDEX <ttt> MODIFY PARTITION <ippp> UNUSABLE;
```

Nach dem Laden können alle Index Partitionen, die zu einer Daten Partition gehören mit dem folgenden Befehl wieder neu aufgenaut warden:

```
ALTER TABLE <ttt> MODIFY <tppp> REBUILD UNUSABLE LOCAL INDEXES;
```

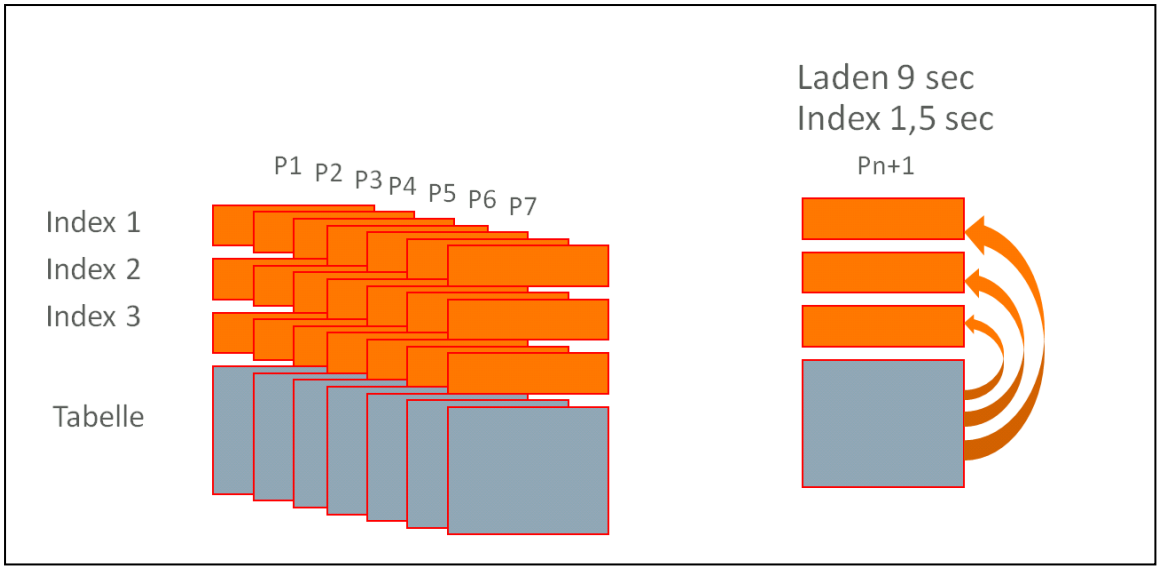

Abb. 5: Wiederaufbau der Indexpartitionen für eine einzelne Partition

Die gesammte Ladezeit beträgt jetzt nur noch 10,5 Sek.

Table Expansion in Version 11.2

Den anderen großen Nachteil, des Index Löschen können wir jetzt mit einem neuen Feature aus der Version 11.2 eliminieren. Das Table Expansion kann unterschiedliche Zugriffspläne für verschiedene Partitionen erzeugen, und die Ergebnisse dann zusammen weiter verarbeiten.

Für die indizierten Partitionen wird ein Index basierter Zugriffsplan verwendet und für die Partitionen mit unusable Index Partitionen wird ein Full Table-Partition Scan genutzt. Auf den größten Teil der Tabelle wird daher mit einem optimalen Zugriffsplan zugegriffen.

Eine Beispiel Query wie die folgende:

```
select count (*) cnt, sum (f.ATR_CTTS) s1
from "/BI0/F0ATR_CR01" F
where KEY_0ATR_CR01P < 65
and   KEY_0ATR_CR01T between 80 and 100
and   KEY_0ATR_CR012 between 6632 and 7792
```

Kann trotz unusable Index Partitionen einen optimalen Zugriffsplan haben, da sie in die folgende Query mit einen Query Rewrite übersetzt wird:

```
Select sum (cnt), sum (s1)
From (select count (*) cnt, sum (f.ATR_CTTS) s1
      from "/BI0/F0ATR_CR01" F
      where PARTITION between 1 and 68
      and   KEY_0ATR_CR01T between 80 and 100
      and   KEY_0ATR_CR012 between 6632 and 7792
      Union all
      select count (*) cnt, sum (f.ATR_CTTS) s1
      from "/BI0/F0ATR_CR01" F
      where PARTITION = 69
      and   KEY_0ATR_CR01T between 80 and 100
      and   KEY_0ATR_CR012 between 6632 and 7792)
```

Für die Partitionen 1 bis 68 wird ein Index Zugriff generiert, während für die Partition 69 ein Full Table-Partition Scan verwendet wird.

```
-------------------------------------------------------------------------
|Id|Operation                                      |Name                |
-------------------------------------------------------------------------
| 0|SELECT STATEMENT                               |                    |
| 1| SORT AGGREGATE                                |                    |
| 2|  VIEW                                         |VW_TE_2             |
| 3|   UNION-ALL                                   |                    |
| 4|    PARTITION RANGE ITERATOR                   |                    |
| 5|     TABLE ACCESS BY LOCAL INDEX ROWID BATCHED|/BI0/F0ATR_CR01    |
| 6|      BITMAP CONVERSION TO ROWIDS              |                    |
| 7|       BITMAP AND                              |                    |
| 8|        BITMAP MERGE                           |                    |
| 9|         BITMAP INDEX RANGE SCAN               |/BI0/F0ATR_CR01~02|
|10|        BITMAP MERGE                           |                    |
|11|         BITMAP INDEX RANGE SCAN               |/BI0/F0ATR_CR01~01|
|12|        BITMAP MERGE                           |                    |
|13|         BITMAP INDEX RANGE SCAN               |/BI0/F0ATR_CR01~04|
|14|    PARTITION RANGE SINGLE                     |                    |
|15|     TABLE ACCESS FULL                         |/BI0/F0ATR_CR01    |
```

Abb. 6: Zugriffsplan mit Table Expansion

Die folgende Tabelle zeigt die jeweiligen Ladezeiten für 1 und für 10 Partitionen:

	1 Request	10 Requests
Ohne Optimierung	100 Sek.	1000 Sek.
Alle Indices löschen	206 Sek.	260 Sek.
Unusable Index Part.	10,5 Sek.	105 Sek,

Tabelle 1: Vergleich der Lade Zeiten

Zu Tableexpansion finden sich weitere Informationen unter:

https://blogs.oracle.com/optimizer/entry/optimizer_transformations_table_expansion

Der Vergleich zeigt deutlich, das die unusable Index Partitionen sowohl für einen einzelnen als auch für 10 Requests deutlich schneller sind. Dabei vermeiden sie jedoch alle Nachteile des vollständigen Index löschen.

Im Hinweis 1842044 ist die Erweiterung für SAP BW beschrieben.

Sie ist in den folgenden Support Packages enthalten:

- SAP NetWeaver BW 7.30 – SP 10
- SAP NetWeaver BW 7.31 – SP 8
- SAP NetWeaver BW 7.40 – SP 3

Falls die geforderten Support Packages noch nicht installiert ist, kann der Hinweis 1842044 auch mit SNOTE eingespielt werden.

Die Funktion wird mit dem folgenden Parameter in der RSADMIN Tabelle eingeschaltet:

OBJECT = ORA_IC_LOAD_NO_DROP VALUE = X

Kontaktadresse:

Jörn Bartels
Oracle Deutschland B.V. & Co. KG
Riesstr. 25
D-80992 München

Telefon: +49 (0) 89-1430 1120
E-Mail Joern.Bartels@oracle.com
Internet: www.oracle.com