

---

# Sind wir eigentlich ganz dicht? - Revisited

Eero Mattila  
Principal Systems Consultant  
[eero.mattila@software.dell.com](mailto:eero.mattila@software.dell.com)

---



Software

# Toad is now a part of Dell.



Software



# Agenda

- Warum überhaupt komprimieren?
- Überblick der Möglichkeiten
- Indexkomprimierung
- Tabellenkomprimierung
- Komprimierung von unstrukturierten Daten
- Hybrid Columnar Compression
- Information Lifecycle Management in 12c
- Backup-Komprimierung
- Fazit



# Warum überhaupt komprimieren?

- Datenexplosion
  - Wir wissen immer mehr...
  - Wir reden noch mehr darüber...
  - Gesetzliche Bestimmungen
- Speicherplatz kostet Geld
- Performance muss stimmen, egal wie viele Daten verarbeitet werden
  - Datenflut ist nicht zu bekämpfen, also müssen wir sie bewältigen
- Manuelle Datenoptimierung ist aufwändig
  - Automatische Datenoptimierung mit Oracle 12c



# Datenverdichtung in Oracle – Überblick

## Oracle interessiert sich dafür – aus gutem Grund

- Index Key Compression seit 8i
- Tabellenkomprimierung seit 9g R2
- Unstrukturierte Daten seit 10g
- Advanced Compression Option seit 11g
- Hybrid Columnar Compression für Exadata V2 und Oracle Storage (Pillar, ZFS) ab 11gR2
- Information Lifecycle Management in 12c
  - Heat Map
  - Automatic Data Optimization



# Indexkomprimierung

- Index Key Compression seit 8.1.5
- Kostenlos verfügbar in allen Editionen
- Non-unique single column Indizes
- Unique und non-unique multi column Indizes
- B\*Tree- Indizes und Index Organized Tables (IOT)
- Transparent für Anwendungen
- Sie haben im Zweifel SEHR viele Indizes!



# Indexkomprimierung (2)

- Eliminierung redundanter Schlüsselwerte in führenden Indexspalten
  - Präfixtabelle im Blockheader, Einträge referenzieren Präfixtabelle
- Kann Platzbedarf und I/O erheblich reduzieren

## Nicht komprimierter Index

```
Hund:ROWID
Katze:ROWID
Maus:ROWID
Hund:ROWID
Hund:ROWID
Maus:ROWID
Katze:ROWID
Maus:ROWID
Hund:ROWID
```

## Komprimierter Index

```
Präfix  0:ROWID
0:Hund  1:ROWID
1:Katze 2:ROWID
2:Maus  0:ROWID
        0:ROWID
        2:ROWID
        1:ROWID
        2:ROWID
        0:ROWID
```

# Indexkomprimierung (3)

- Definition beim Erstellen oder auch nachträglich:

```
CREATE INDEX index ON tabelle (spalte1, ...) COMPRESS x;
```

```
ALTER INDEX index REBUILD [ONLINE] COMPRESS x;
```

x = Anzahl der zu komprimierenden führenden Spalten





# Indexkomprimierung (4)

- Ermitteln der optimalen Komprimierungsrate

```
ANALYZE INDEX comp_test VALIDATE STRUCTURE;
```

```
SELECT blocks, opt_cmpr_count, opt_cmpr_pctsave  
FROM index_stats;
```

```
BLOCKS OPT_CMPR_COUNT OPT_CMPR_PCTSAVE  
-----  
988 2 37
```

OPT\_CMPR\_COUNT gibt die Anzahl der zu komprimierenden Spalten an,  
OPT\_CMPR\_PCTSAVE die zu erwartende Platzersparnis in %



# Indexkomprimierung (5)

- Probe aufs Exempel

```
ALTER INDEX comp_test REBUILD COMPRESS 2;  
ANALYZE INDEX comp_test VALIDATE STRUCTURE;
```

```
SELECT blocks, opt_cmpr_count, opt_cmpr_pctsave  
FROM index_stats;
```

```
BLOCKS OPT_CMPR_COUNT OPT_CMPR_PCTSAVE  
-----  
621                2                0
```

- Vorsicht mit ANALYZE INDEX ... VALIDATE STRUCTURE
  - DML-Lock!
  - Es gibt zwar auch die Option ONLINE, die aber keine Stats sammelt...



# Tabellenkomprimierung

- Eingeführt mit Oracle9i R2 für Bulk-Load-Operationen
  - Direct Path Load, CTAS, paralleles DML, Insert APPEND
  - Nur interessant für Datawarehouse – und auch dort nicht immer brauchbar
- Ab Oracle 11g: Advanced Compression Option
  - Separate Lizenz
  - Komprimierung für OLTP
    - Neuer Name in 12c: Advanced Row Compression
  - Basic Compression (Bulk-Operationen) weiterhin kostenlos
- Komprimierung auf Blockebene
  - Wiederkehrende Spaltenwerte in Symboltabelle im Blockheader
    - Auch spaltenübergreifbar
  - Ersparnis bei Speicherplatz sowie logischem und physikalischem I/O



# Tabellenkomprimierung (2)

- Syntax:

```
CREATE TABLE [...] COMPRESS FOR [OLTP|BASIC];
```

```
ALTER TABLE tabelle COMPRESS FOR [OLTP|BASIC];
```

(Statt OLTP geht auch ALL OPERATIONS)

## Ab Oracle 12c auch:

```
CREATE|ALTER TABLE [...] ROW STORE COMPRESS ADVANCED;
```

Einzigste erlaubte Syntax bei ILM-Policies (s. später)



# Tabellenkomprimierung (3)

- Vorhandene volle Blöcke werden nicht komprimiert
  - Komprimierung greift erst, wenn Sätze gelöscht und eingefügt werden
  - Wenn die Tabelle gleichzeitig reorganisiert wird, greift Komprimierung sofort:

```
ALTER TABLE tabelle MOVE COMPRESS FOR [OLTP|BASIC]; -- offline!  
Oder DBMS_REDEFINITION (online)
```

- Standardeinstellung für Tablespace

```
CREATE|ALTER TABLESPACE [...] DEFAULT COMPRESS FOR [OLTP|BASIC];
```



# Tabellenkomprimierung (4)

- Beispiel:

```
CREATE TABLE doag_normal  
AS SELECT * FROM dba_objects;
```

```
CREATE TABLE doag_compr COMPRESS FOR ALL OPERATIONS  
AS SELECT * FROM dba_objects;
```

```
SELECT segment_name, blocks  
FROM user_segments  
WHERE segment_name LIKE 'DOAG%';
```

SEGMENT_NAME	BLOCKS
DOAG_COMPR	<b>384</b>
DOAG_NORMAL	<b>1152</b>



# Tabellenkomprimierung (5)

- Welche Tabellen komprimieren?
  - Oracle-Empfehlung: Alles (außer LONG, Queue-Tabellen) – hmm?
  - Große Tabellen mit wiederkehrenden Werten
  - Größere Blöcke ermöglichen höhere Verdichtung
  - 80/20-Regel: 20% der Tabellen machen 80% des Speicherplatzes aus
- 11g R2: DBMS\_COMPRESSION.GET\_COMPRESSION\_RATIO
- Früher: DBMS\_COMP\_ADVISOR (auf OTN verfügbar)
- Nur BASIC, OLTP, HCC
  - Keine Unterstützung für Indizes, unstrukturierte Daten
- Automatic Segment Advisor
  - Ab 11g R2 Empfehlungen für Komprimierung
  - Diagnostic und Tuning Packs zusätzlich erforderlich



# Was kostet die Komprimierung?

- Advanced Compression Option ist lizenzpflichtig
- Keine Zusatzinstallation – immer da in Enterprise Edition
  - Zauberwort COMPRESS genügt...
- Hidden Parameter `_OLTP_COMPRESSION=FALSE`
  - Zieht nicht mehr?
- CPU statt I/O und Speicherplatz?
  - Kein messbarer Overhead beim Lesen
  - Performance-Steigerung durch geringeren I/O (weniger Blöcke)
  - Meist geringfügiger Overhead bei DML -> **Testen!**
    - Ein „Hot Block“ kann noch „heißer“ werden, weil mehr Einträge pro Block!
  - Asynchrone Komprimierung, wenn bestimmter Blockfüllgrad erreicht
- Transparent für Anwendungen





# Komprimierung unstrukturierter Daten

- Oft ideale Kandidaten, wenn nicht per se komprimiert
- Oracle10g: UTL\_COMPRESS für RAW, BLOB, BFILE
- SecureFiles ab Oracle11g
  - Setzt ACO voraus
  - Komprimierung in 3 Stufen: LOW, MEDIUM, HIGH
    - Industriestandard-Algorithmen
  - Deduplizierung
    - Identische Dateien werden nur einmal gespeichert
  - CPU-Overhead vs. Platzersparnis -> **Testen!**



# Hybrid Columnar Compression

- Exadata V2, Pillar Axiom, Sun ZFS Storage Appliance
- Logische Komprimierungsgruppen (Compression Unit)
  - Umfasst mehrere Blöcke -> größeres Komprimierungspotenzial
  - Jede **Spalte** wird einzeln komprimiert
  - Wiederkehrende Daten in gleichen Blöcken
- Optimiert für Warehouse- oder Archiv-Systeme
  - `COMPRESS FOR QUERY [LOW|HIGH]`    --Abfrage-Performance
  - `COMPRESS FOR ARCHIVE [LOW|HIGH]`    --Platzersparnis
- Sehr hohe Komprimierung bei optimaler Performance
  - Faktor 15: „aus 1 TB werden ~67 GB“ – **testen!**




# Information Lifecycle Management in 12c

- Ziel: Selten verwendete Daten **automatisch** komprimieren und/oder auf preiswertere Medien auslagern
- ACO muss lizenziert sein
- Bestandteile:
  - Heat Map
  - Automatic Data Optimization



# Heat Map

- Zeichnet auf Block-, Segment- und Partitionsebene auf, wie oft auf Daten lesend und/oder schreibend zugegriffen wird
  - „heiße“, „warme“, „kalte“ Daten
- Unterscheidet zwischen Lesen, Schreiben, FTS und Index-Zugriff; ignoriert administrative Zugriffe (Stats Gathering, DDL, Redefinition etc.)
- Einschalten über init.ora-Parameter
  - `ALTER SYSTEM SET HEAT_MAP=ON;`
- ORA-38342: Wärmebild nicht aktiviert 

# Heat Map (2)

```
select owner, object_name,  
       to_char(segment_write_time, 'dd.mm.yy hh24:mi:ss') as write,  
       to_char(segment_read_time, 'dd.mm.yy hh24:mi:ss') as read,  
       to_char(full_scan, 'dd.mm.yy hh24:mi:ss') as fts,  
       to_char(lookup_scan, 'dd.mm.yy hh24:mi:ss') as lookup  
from dba_heat_map_segment where owner = 'DEMO';
```

OWNER	OBJECT_NAME	WRITE	READ	FTS	LOOKUP
DEMO	DOAG_NORMAL	17.11.13 17:56:14		17.11.13 17:56:14	17.11.13 17:56:14
DEMO	DOAG_COMPR			17.11.13 17:56:14	
DEMO	MIST			17.11.13 20:56:18	
DEMO	PK_DOAG_NORMAL				17.11.13 17:56:14
DEMO	PK_DOAG_COMPR				17.11.13 17:56:14
DEMO	EMP	17.11.13 22:56:21		17.11.13 22:56:21	
DEMO	TEST	18.11.13 02:22:23		18.11.13 02:22:23	



# Automatic Data Optimization

- Komprimiert und/oder verschiebt alte oder „kalte“ Daten
- Steuerung über Richtlinien (Policies)
  - Zeile
  - Segment
  - Tablespace (Storage Tier)
- Richtlinien basieren auf
  - Zeit: Tage, Monate, Jahre
  - Zugriffshäufigkeit: kein Zugriff, wenige Zugriffe, keine Änderungen, seit Erstellung
  - Benutzerdefinierte Funktion
  - Tablespace: Freier Speicherplatz, definiert über ILM-Parameter
- Auswertung und Ausführung der Richtlinien
  - automatisch im Wartungsfenster
  - manuell



# Automatic Data Optimization (2)

- Beispiele:

```
CREATE TABLE [...] ILM ADD POLICY  
ROW STORE COMPRESS ADVANCED ROW -- blockweise Komprimierung  
AFTER 3 DAYS OF NO MODIFICATION; -- keine DML-Zugriffe
```

```
ALTER TABLE [...] ILM ADD POLICY  
COMPRESS SEGMENT -- ganze Tabelle komprimieren  
AFTER 30 DAYS OF NO ACCESS; -- kein Zugriff
```

```
ALTER TABLE [...] ILM ADD POLICY -- auslagern  
TIER TO tablespace_auf_billigem_speichermedium; -- wenn TBS „voll“
```



# Backup-Komprimierung

- RMAN

- Blockweise, physikalische Datensicherung

```
CONFIGURE DEVICE TYPE [DISK|SBT] BACKUP TYPE TO COMPRESSED  
BACKUPSET;
```

```
CONFIGURE COMPRESSION ALGORITHM '[BASIC|LOW|MEDIUM|HIGH]'
```

- BASIC kostenfrei, sonst Advanced Compression Option

- Data Pump Export

- „Logisches“ Backup
- Parameter `COMPRESSION [ALL|METADATA_ONLY|DATA_ONLY]`
- ACO erforderlich





# Backup-Komprimierung (2)

- Dekomprimierung bei Restore bzw. Import transparent
- Performance abhängig von vielen Faktoren
  - Charakter der Daten, Netzwerk, CPU-Ressourcen usw.
- Gründlich testen, vor allem Restore!
  - Was nützt die kleinste Sicherung, wenn die Wiederherstellung ewig dauert...



# Fazit

- Komprimierung kann gute Dienste leisten
- Der Möglichkeiten sind viele
  - Index, Tabelle, unstrukturierte Daten, Hybrid, Backup
  - Oracle 12c mit Heat Map und ADO: Automatisierung!
- Vorsicht – meist Extrakosten (ACO)
- Meistens Performance-Vorteile
  - aber auch Nachteile möglich
- **Gründlich testen!**



# Gewinnen Sie ein Dell XPS™ Tablet im Wert von 299€



und entdecken Sie Dell Software für Qualität,  
Performance und Verfügbarkeit Ihrer Datenbanken.

Verlosung am Donnerstag, 21.11.2013 nach der  
Keynote um 14:50 Uhr am Dell Software Stand!

Herzlichen Dank für Ihre Aufmerksamkeit!

Welche Fragen haben Sie?

[eero.mattila@software.dell.com](mailto:eero.mattila@software.dell.com)

