



Compression

Doag Konferenz 2013

Dr. Günter Unbescheid
Database Consult GmbH
Jachenau

Database Consult GmbH

- Gegründet 1996
- Kompetenzen im Umfeld von ORACLE-basierten Systemen
- Tätigkeitsbereiche
 - Security, Identity Management
 - Tuning, Installation, Konfiguration, Systemanalysen
 - Support, Troubleshooting, DBA-Aufgaben
 - Datenbankdesign, Datenmodellierung und –design
 - Maßgeschneiderte Workshops
 - www.database-consult.de
- Seit 2012 – Kooperation mit



Motivation und Ausblick

- Offensichtlich: Reduktion des Daten- /Speichervolumens
- Reduziert
 - Memory-/Cache-Anforderungen
 - IO-Anforderungen (logical/physical) – mit Ausnahmen
- Erhöht
 - CPU-Anforderungen – mit Ausnahmen
 - Lizenzkosten – mit Ausnahmen
 - Block-Contention bei Veränderungen
- Abhängigkeiten
 - Kompressionsraten stark abhängig von DML-Charakteristik
 - ... von Blockgrößen, ggf. von Sortierungen
- Daher sorgfältige Planung und Tests (De-Sign oder Nicht-Sein)



Überblick und Lizenzierung

Feature	ab	Load	SE	EE	Option
Index Key Compression	8i	--	N	Y	keine
Basic Table Compression	9i	direct	N	Y	keine
OLTP Table Compression	11g R1	all	N	Y	ACO
Hybrid Columnar Compr.	11gR2	direct	N	Y	Exadata, ZFS oder Pillar Axiom 600 storage
RMAN	11gR1	--		Y	ACO (Ausnahme BASIC)
ADO/Heat Map	12c	all	N	Y	ACO
Secure Files	11gR1	All	N	Y	ACO
Data Pump	11g R1	--	N	Y	ACO
Data Guard Redo	11gR2	--	N	Y	ACO
Network	12c	--	N	Y	ACO
Flashback Data Archive	12c	--	N	Y	ACO



Hintergrund

- Kompressionsraten sind abhängig von Daten- und Lade- und Änderungscharakteristiken
- Metadaten liefern nur die Segment-Definitionen
- Genaue Beurteilung oftmals nur durch zusätzliche Storage- und Block-Analysen:
`dbms_space.space_usage`, `dbms_compression`
- Zusätzliche Hilfsmittel über Metriken, z.B. `HSC OLTP Space Saving`
- `analyze index ... validate structure (index_stats)`

```
alter session set tracefile_identifizier = <string>
alter system checkpoint;
alter system dump datafile <fno> block <bno>;

oradebug SETMYPID;
oradebug TRACEFILE_NAME;
```



Basic Compression

Tabellen für direct loads



Block-Dump ohne Compression 11g

```
Block dump from cache:
>>
Object id on Block? Y
seg/obj: 0x1577e csc: 0x00.d5e797 itc: 2 flg: E typ: 1 - DATA
>>
ntab=1
nrow=276
>>
0xe:pti[0] nrow=276 offs=0
0x12:pri[0] offs=0x1f83
0x14:pri[1] offs=0x1f6e
>>
tab 0, row 0, @0x1f83
tl: 21 fb: --H-FL-- lb: 0x0 cc: 3
col 0: [ 2] c1 02
col 1: [ 6] 61 61 62 62 63 63
col 2: [ 7] 78 71 09 1b 01 01 01
>>
tab 0, row 1, @0x1f6e
tl: 21 fb: --H-FL-- lb: 0x0 cc: 3
col 0: [ 2] c1 03
col 1: [ 6] 61 61 62 62 63 63
col 2: [ 7] 78 71 09 1b 01 01 01
```

The diagram consists of four blue boxes on the right side, each with an arrow pointing to a specific part of the dump:

- Data Object ID (DBA_OBJECTS)**: Points to the **seg/obj: 0x1577e** field.
- Table/row Directory**: Points to the **ntab=1** and **nrow=276** fields.
- Table/row Index**: Points to the **0xe:pti[0] nrow=276 offs=0** field.
- Row Eintrag**: Points to the **tab 0, row 1, @0x1f6e** field.



Dump Basic Compression

```
ntab=2  
nrow=719
```

2 „Tabellen“

```
---  
0x1e:pti[0]      nrow=6  ofs=0  
0x22:pti[1]      nrow=713  ofs=6  
---
```

```
tab 0, row 0, @0x1f3b  
tl: 13 fb: --H-FL-- lb: 0x0  cc: 2  
col 0: [ 7] 78 71 09 1b 10 2e 29  
col 1: [ 8] 54 65 73 74 74 65 78 74  
bindmp: 01 3e 02 03 d0 54 65 73 74 74 65 78 74
```

Symbol Table (2 Spalten)
3e – Referenzierungen im Block

```
tab 1, row 0, @0x1f32  
tl: 9 fb: --H-FL-- lb: 0x0  cc: 3  
col 0: [ 7] 78 71 09 1b 01 01 01  
col 1: [ 6] 61 61 62 62 63 63  
col 2: [ 2] c1 02  
bindmp: 2c 00 03 01 04 05 ca c1 02
```

Geänderte Column Order
Referenzierung bindmp
Reduzierte tl

```
perm_9ir2[3]={ 2 1 0 }
```



Dump Basic Compression

tab 1, row 1, @0x1f1a (zweite "wirkliche" Row)

```
tl: 16 fb: --H-FL-- lb: 0x0 cc: 3
col 0: [10] 58 58 31 58 58 58 58 58 58 58 (XX1XXXXXXXX)
col 1: [ 6] 32 41 41 41 41 41 (2AAAAA)
col 2: [10] 32 5a 5a 5a 5a 5a 5a 5a 5a 5a (2ZZZZZZZZZZ)
bindmp: 2c 00 02 03 00 c9 32 5a 5a 5a 5a 5a 5a 5a 5a
```

tab 0, row 0, @0x1f45 (Symbol Row A)

```
tl: 11 fb: --H-FL-- lb: 0x0 cc: 2
col 0: [10] 58 58 31 58 58 58 58 58 58 58 (XX1XXXXXXXX)
col 1: [ 6] 32 41 41 41 41 41 (2AAAAA)
bindmp: 00 6e 02 05 ce 32 41 41 41 41 41
```

tab 0, row 5, @0x1f66 (Symbol Row B)

```
tl: 13 fb: --H-FL-- lb: 0x0 cc: 1
col 0: [10] 58 58 31 58 58 58 58 58 58 58
bindmp: 00 02 d2 58 58 31 58 58 58 58 58 58 58
```



Basic Compression Metadaten

C1	NOT NULL	NUMBER
C2		VARCHAR2 (50)
C3		DATE

TABLE_N	COMPRESSION	COMPRESS_FOR	PCT_FREE	BLOCKS	NUM_ROWS	AVG_ROW_LEN
T0	DISABLED		10	5	995	21
T1B	ENABLED	BASIC	0	5	995	21

NOCOMPRESS

BL	FIL	NUM_ROWS
131	8	276
133	8	279
134	8	279
135	8	161

COMPRESS BASIC

BL	FIL	NUM_ROWS
203	8	713
204	8	282



Basic Compression

- Keine Binärkompression sondern De-Duplizierung
- Unit ist der betreffende Oracle-Block
- De-Duplizierung nur bei direct load/append
- Zusätzliche und notwendige Kriterien
 - Wiederholgruppen pro Block
 - Ausreichende Füllung
- Umsortierung der Columns innerhalb eines Blockes - effizient
- Verhalten bei update:
 - Spalte ohne Symbol-Referenz: Entpacken, neuer Offset im Block oder neuer Block (Migration)
 - Spalte mit Symbol-Referenz: Entpacken und ggf. Migration
 - Verhalten bei konventionellem Insert – keine Komprimierung



Sortierung, Blockgröße ...

- Blockgröße
 - Je größer die Datenblöcke, desto höher die Effizienz bei gegebener Row-Länge für die De-Duplizierung
- Sortierung
 - Die Auswirkungen der Row-Sortierung werden durch die Umschichtung der Columns innerhalb der Blöcke minimiert
 - Signifikante Auswirkungen nur bei „kleineren“ Blöcken vs. „längeren“ Rows
- Beim Laden ggf. FK-Constraints ausschalten, um direct load zu ermöglichen
- Keine add/drop column Aktionen – schema modifications
- Maximale Anzahl Spalten - 255



Advanced Compression

Tabellen für OLTP



Advanced Table Compression

- Klauseln
 - **compress for oltp** (11g)
 - **row store compress advanced** (12c)
- Gleiches Block-Format wie Basic Compression – De-Duplizierung
- Wirksam bei konventionellen **insert**-Operationen, wenn
 - Pctfree-Grenze erreicht wird
- Keine Komprimierung bei Updates
 - Auch dann nicht, wenn Füllgrenze erreicht wird
 - In diesem Fall: Migration vor Kompression
- Umladung empfohlen über direct load
- Gemischte Kompressions-Stati bei Blöcken möglich
- Effizienz daher stark abhängig von DML-Charakteristik
- Erhöhtes REDO/UNDO vor Batch-Komprimierungen



Advanced Table Compression

- „Rollup“-Effekte bei herkömmlicher Grundfüllung
 - Optimieren durch direct load
- Weitere Effekte:
 - Spontanität der OLTP-Transaktionen (unsortiert) – Zufälligkeit bei Datenredundanzen
 - Freilisten oder ASSM weisen Blöcke nach Freiplatz und Instanz-Affinität und nicht nach Datencharakteristik zu
 - Ggf. Buffer busy Effekte wegen höherer Blockfüllung
 - Effizienz kann/muss nur durch Tests ermittelt werden



Advanced Compression Advisor

Nützliches Werkzeug zur Kalkulation



Advanced Compression Advisor

- Abschätzung von Kompressionsraten
- Ausgabe von Compression Infos für Rowid
- Bordmittel ab Version 11g R2 – **dbms_compression**
- Separater Download für 9iR2 bis 11g R1 – **dbms_comp_advisor**
- Hinreichend präzise Abschätzung
- Interessant auch für nicht lizenzierte Kompressionsmodelle
- Scratch Tablespace für temporäre Daten nötig – kann angegeben werden
- Ab 12c auch für Secure Files geeignet
- Für Indizes alternativ: `analyze index ... validate structure`
 - `index_stats`: **OPT_CMPR_COUNT** **OPT_CMPR_PCTSAVE**



Advanced Compression Advisor

```
exec DBMS_COMPRESSION.GET_COMPRESSION_RATIO . . .
```

```
0.880 secs elapsed.
```

```
1.220 secs elapsed.
```

```
BLKCNT_CMP = 410
```

```
BLKCNT_UNCMP = 1058
```

```
ROW_CMP = 180
```

```
ROW_UNCMP = 69
```

```
CMP_RATIO = 2.5
```

```
COMPTYPE_STR = "Compress For OLTP,"
```

```
exec DBMS_COMPRESSION.GET_COMPRESSION_TYPE // ROWID //
```

Error



ORA-20000: Compression Advisor must have at least 1000000 rows in this table/partition segment (segment has 74201 rows)
ORA-06512: at "SYS.PRVT_COMPRESSION", line 776
ORA-06512: at "SYS.DBMS_COMPRESSION", line 214
ORA-06512: at line 25



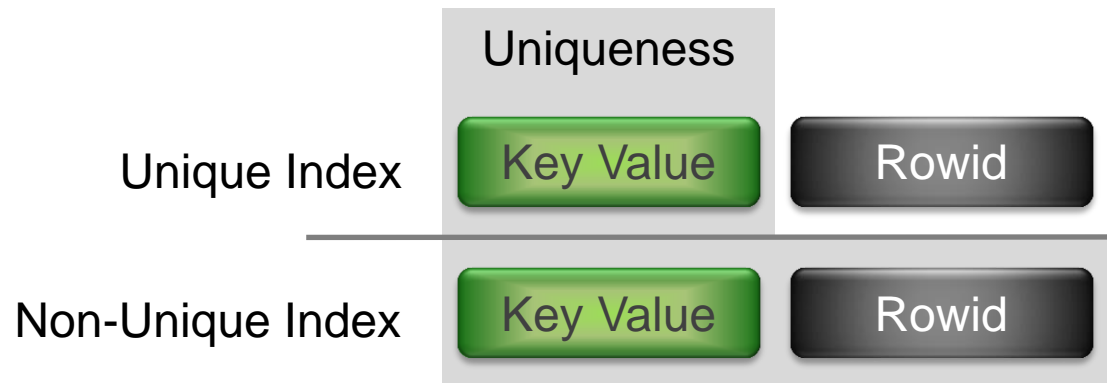
Index Compression

Key Compression



Index Compression

- Verfügbar für B-Tree Indizes und IOTs – dort für LEAVE-Blöcke
- Definition von Prefix Spalten
 - Prefixe werden aus distinktiven Wertekombinationen gebildet
 - Daher Unterschied zu Table Compression
- Prefix Minimum/Maximum
 - Unique: Anzahl Indexspalten -1, sonst Anzahl Indexspalten
 - Maximum ist nicht immer das Optimum,
- Compression als De-Duplizierung – in „real time“
- Compression Unit ist der Datenblock



Index Compression – non-unique

INDEX_NAME	COMPRESSION	PREFIX_LENGTH	PCT_FREE	NUM_ROWS
T1B_IDXC	ENABLED	1	10	995

```
----- begin tree dump
branch: 0x2000093 33554579 (0: nrow: 2, level: 1)
  leaf: 0x2000094 33554580 (-1: nrow: 649 rrow: 649)
  leaf: 0x2000095 33554581 (0: nrow: 346 rrow: 346)
----- end tree dump
```

Anzahl
Leaf Rows

NC >> 578
C >> 649

```
kdxcoopc 0xa0: opcode=0: iot flags=-C- is converted=Y
kdxconro 649
kdxledsz 0
prefix row#0[8021] flag: -P----, lock: 0, len=11
col 0; len 8; (8): 54 65 73 74 74 65 78 74
prc 600 >>>
prefix row#1[2609] >>> prc 49
>>>
row#0[8012] flag: -----, lock: 0, len=9
col 0; len 6; (6): 02 00 00 cb 01 8b
psno 0
```



Index Compression - Diverses

- Keine De-Duplizierung über Spalten hinweg
- Column Order bei zusammengesetzten Indizes
 - Ohne Compression – höchst selektive Spalte vorne
 - Compression – Sortierung nach Redundanz, Auswirkung bei partieller Nutzung
- IOTs wie unique Indizes – Compression nur ab 2 Spalten

Identische Werte in unterschiedlichen Columns:

```
prefix row#0[8008] flag: -P----, lock: 0, len=24
col 0; len 10; (10):  31 58 58 58 58 58 58 58 58 58
col 1; len 10; (10):  31 58 58 58 58 58 58 58 58 58
```

TABLE_NAME	COMPRESSION	COMPRESS_FOR
TIOT	DISABLED	BASIC

TABLE_NAME	INDEX_TYPE	COMPRESS
TIOT	IOT - TOP	ENABLED



Large Objects

LOB und Secure Files Compression



Compression Large Objects

- Basicfile – Traditionelle LOB Storage
 - Keine Kompressionsmöglichkeiten des LOB
 - De-Duplizierung für die restliche Row möglich, aber wenig wirksam
enable/disable storage in row
 - Beachten: bei Multibyte CS der DB wird LOB UCS-2 kompatibel gespeichert: doppelte Char-Länge!
- Securefiles
 - Parameter **db_securefile** – **permitted** (11g) / **preferred** (12c)
 - Binäre Komprimierung nach Industriestandards , z.B. GZIP
 - Unabhängig von OLTP/BASIC
 - Dynamisch, keine Komprimierung von bereits komprimierten Daten



Secure Files

- Unterschiedliche Kompressionsstrategien
 - **low, medium, high**
- Balance zwischen Kompressionsrate und CPU-Verbrauch
- Vergleich mit XML-File als CLOB – Ladung
 - **dbms_space.space_usage** (lob segment name, used_bytes)
 - **CPU used by this session, undo change vector size**
 - **securefile compressed bytes, securefile ***

	Datei	Basicfile CLOB	SecureFile CLOB	SF low CLOB	SF medium CLOB	SF high CLOB
Bytegröße 1 LOB	34018	73728	73728	16384	8192	8192
Bytes komplett		9.437.184	9.510.912	2.113.536	1.056.768	1.056.768
CPU		17	22	12	19	29
redo		328248	445304	202936	131432	131428
undo cv		89.648	167.176	75.404	42.428	42.428
elapsed t.		01.027605	00.571199	00.258117	00.276451	00.362111



Hybrid Columnar Compression

Exadata, ZFS und Pillar Storage

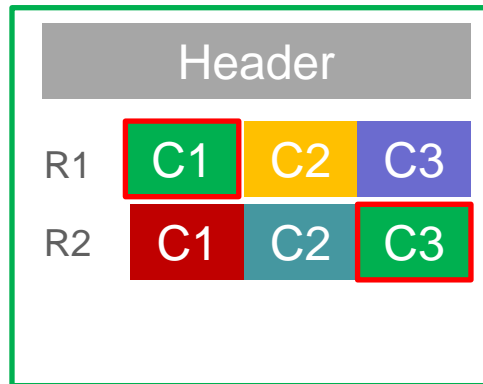


- Ursprünglich im Exadata Umfeld, mittlerweile auch für
 - ZFS Storage Appliances (über NFS)
 - Pillar Axiom Systeme (über Fibre Channel), SPAR Super Cluster
- Nutzung eines neuen „hybriden“ Blockformats
- Bessere Effizienz durch
 - vergrößerte Kompressionseinheiten
 - Interner Sortierung der Spalten
- Ausprägungen: Query und Archive mit Optionen low und high
 - Balance zwischen CPU-aufwand und Kompressionsraten
 - Unterschiedliche algorithmen; LZO (4x), ZLIB (6-7x), bzip2 (12x)
- Unterstützt bei **direct load/append** Operationen
- Fallback Methode – OLTP Compression
 - Non-direct path load, updates

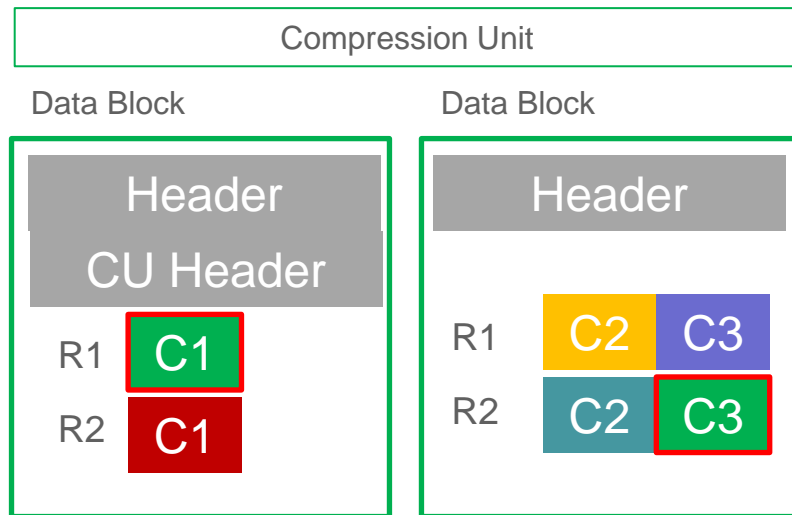


HCC

Data Block
Compression Unit



Row Format



Hybrid Format
Variable CUs



```
BH (0x259dfc518) file#: 57 rdba: 0x0e648c4b
(57/2395211) class: 1 ba: 0x258a5e000
ntab=1
nrow=1
tab 0, row 0, @0x30
tl: 8016 fb: --H-F--N lb: 0x0 cc: 1
nrid: 0x0e648c4c.0
col 0: [8004]
Compression level: 03 (Archive Low)
Length of CU row: 8004
CU total length: 10092
ncols: 5
nrows: 16384
num deleted rows: 0
START_CU:
BH (0x12fe06118) file#: 57 rdba: 0x0e648c4c
(57/2395212) class: 1 ba: 0x12eb2e000
tab 0, row 0, @0x1734
tl: 2124 fb: -----LP- lb: 0x0 cc: 1
tab 0, row 1, @0x32
tl: 5890 fb: --H-F--N lb: 0x0 cc: 1
nrid: 0x0e648c4d.0
CU total length: 19873
```



HCC - Besonderheiten

- Support auch für RMAN und Data Guard
- Schema Evolution (add/drop columns)
- Transfer auf NON-HCC: Dekomprimieren vor dem Zugriff
- Locking der gesamten CU bei update einer beliebigen Row
- Ideal für „full“ Scans
 - Smart scan – offloading von decompression
 - hängt ab von serial direct path reads (non-parallized operations) („große“ Objekte)
 - Beachten bei Partitionen (Größe)
 - in-memory parallel query
 - Daten gehen ins Buffer Cache
 - keine Nutzung Serial Direct IO daher kein Smart Scan



- Smart Scan Ablauf:
 - A CU is buffered => Predicates processing =>
 - Predicate columns decompressed => Predicate evaluation =>
 - Reject CU's if no row satisfies the predicate =>
 - For satisfying rows, the projected columns are decompressed =>
 - A small CU is created with only projected and predicate columns =>
 - Returned to the DB server.



- Single Row Access, z.B. per Index
 - Lesen der gesamten CU – erhöhte IO-Raten
 - Decompression/CPU Load auf DB-server
- HCC könnte auf Grund von Größenreduzierung SDPR ausschalten
 - Zusätzliche CPU-Last auf dem DB-Server
- Konventioneller Update
 - Row zieht um in OLTP Compression Block (neue RowId)
 - Dort wird sie – zunächst – nicht komprimiert
 - Erreichbar über alte und neue RowId
 - CU der migrierten Row zeigt delete flog
- Neue CUs nur nach direct load / append



HCC

	NC	BC	OC	QL	QH	AL	AH
Rows	1 Million	1 Million	1 Million	1 Million	1 Million	1 Million	1 Million
Load Time	04.087942	04.262559	07.529916	03.923452	05.369683	06.877911	13.077727
CPU	420	432	741	421	565	720	1330
redo	135172616	97018232	366862524	8591652	781288	772884	272564
Undo cv	4084216	46196	223286124	12816	7312	7312	4116
IDL blocks	0	11766	0	0	0	0	0
OLTP blocks	0	0	13136	0	0	0	0
Cus compressed	0	0	0	281	35	36	31
Used blocks	16.391	11.764	13.157	1.037	91	90	31



Neuigkeiten und 12c

ADO und Heat Maps, Network etc.



Automatic Data Optimization (ADO)

- Policies steuern Tiering und Kompressionsaktionen automatisiert über Änderungscharakteristiken von Segmenten
- Heat Maps als Fortführung des Table Monitoring
 - DBA_HEAT_MAP_SEG_HISTOGRAM/SEGMENT
 - Package DBMS_HEAT_MAP
 - Init.ora HEAT_MAP = ON (non default)
- Policy
 - Über create/alter table
 - Zeitraum: Tage, Monate, Jahre
 - Aktion: keine / gelegentliche Veränderungen / Zugriffe
 - Kontext: Segment, Block oder Tablespace
 - Komprimierungsart: eine HCC oder OLTP

```
OWNER  
OBJECT_NAME  
SUBOBJECT_NAME  
TRACK_TIME  
SEGMENT_WRITE  
FULL_SCAN  
LOOKUP_SCAN
```



- Advanced Network Compression – ANC Option
 - Über Netzprofil (sqlnet.ora) oder Alias (tnsnames.ora)
 - Einstellungen high oder low sowie minimale Datengröße (threshold)
- RMAN – ANC (ausser basic)
 - Basic, low, medium und high
 - Precompression processing – Optimierung des Freiplatzes in Blöcken (Vorsicht!)
 - Auch auf compressed tables
- Data Guard redo Transport Compression
- In-database Archiving
 - Markierung von Rows (inactive)



Danke für's Zuhören
www.database-consult.de

