

Compression und Mining Proof of Concept bei Siemens Healthcare

Heinz Mielimonka
ORACLE Deutschland B.V. & Co. KG
Nürnberg

Schlüsselworte:

Compression

Einleitung

Bei der Remote Service Platform des Siemens Konzerns fallen sehr große Datenmengen an. Diese zu reduzieren, aber gleichzeitig die Analysequalität zu steigern war erklärtes Projektziel. Dazu wurde in Zusammenarbeit mit Oracle von Siemens Healthcare, dem Betreiber der Platform, ein Proof of Concept (PoC) aufgesetzt. Hierbei wurde eine repräsentative Menge von realen Daten der Remote Service Platform in ein Oracle Datenbank gebracht und mit verschiedenen Kompressionsverfahren komprimiert. Hr.Mielimonka, Projektleiter Oracle und Hr.Pyka, Projektleiter von Siemens, berichten hier über den Projektverlauf und die Ergebnisse.

Remote Service Platform

Mit der Remote Service Platform werden bei Siemens entfernte Systeme z.B. die Antriebstechnik der AIDA Flotte (siehe Abb. 1) überwacht und gewartet. Zum April 2011 sind 145.000 weitere Systeme wie Computertomographen, Gasturbinen bis hin zu Walzwerken an dieses System angeschlossen. Dabei werden die anfallenden Betriebsdaten der entfernten Systeme über einen sicheren SSL VPN Tunnel in die Siemens DMZ geleitet. Von dort gelangen sie über mehrere Firewalls in die Remote Service Platform und werden dort als komprimiertes ASCII File abgelegt. Zur Analyse werden diese Daten entkomprimiert und in die Analyseumgebung geladen. Die heutige Datenbankplatform der Remote Service Platform ist Oracle.



Abb. 1: Remote Services für AIDA Cruises
Quelle: [4. April 2011 Pressekonferenz von Prof. Dr. Siegfried Russwurm auf der Hannover Messe 2011](#)

Der Erfolg und das von Siemens in diesem Bereich erwartete Datenwachstum führen dazu, dass die Betriebskosten immer weiter stiegen. Eine zentrale Rolle innerhalb der Betriebskosten nehmen die Storagekosten ein. Aus diesem Grund hat sich Siemens an Oracle gewandt um Lösungskonzepte gemeinsam zu erarbeiten. Als ersten Schritt wurden die Kompressionsmöglichkeiten innerhalb der Oracle Datenbank genauer untersucht.

PoC Vorbereitung

Siemens Healthcare stellte die notwendige Hardware (PRIMERGY TX300 S5 mit 16 CPUs und 32 GB RAM) und ein 40GB großes repräsentatives Datenset für die Tests zur Verfügung. Nach der Installation der Oracle Datenbank 11.2.0.2 Enterprise Edition auf Microsoft Windows Server 2003 R2 Server (64bit), standen wir bereits vor der ersten Herausforderung. Tausende von ASCII Files waren auf hunderte von Unterverzeichnissen verteilt, da ja jedes remote system einen eigenen Bereich, der nach Zeit strukturiert ist, befüllt. Diese verteilten Daten mit dem Oracle Loader einzulesen war mit dem vorbereiteten Script nicht möglich.

```
LOAD DATA
INFILE      '...\data\file01.txt'
BADFILE     '...\file01.bad'
DISCARDFILE '...\file01.dsc'
APPEND
INTO TABLE testhc.hc_table_normal
FIELDS TERMINATED BY 'X'9'
(
  id          SEQUENCE (MAX,1) ,
  flag       ,
  log_date   ,
  log_time   ,
  component1 ,
  component2 ,
  log_message CHAR(4000)
)
```

Allerdings konnte ein versierter Siemenskollege das von mir vorbereitete Script „dynamisieren“ und uns so in die Lage versetzen, quasi auf Knopfdruck die Daten einzulesen. Es war geplant die Daten in einen (unkomprimierten) Tablespace hc_normal in die Tabelle HC_TABLE_NORMAL einzulesen.

```
CREATE TABLESPACE hc_normal DATAFILE '...\hc_normal.dbf' SIZE 10G REUSE
AUTOEXTEND ON NEXT 1G MAXSIZE UNLIMITED DEFAULT NOCOMPRESS;
```

Während des unbeaufsichtigten Ladeprozesses lief der Tablespace voll. Nachdem wir den Tablespace erweitert habe konnte das Datenladen fortgesetzt werden. Am Ende hatten wir 196.497.912 Datensätze geladen. Der Tablespace belegte 40.067 MB Speicherplatz.

Die beiden für die Komprimierung vorbereiteten Tablespaces hc_compress1 und hc_compress2 waren noch leer. Entsprechende Tabellen hc_table_compress_1 und hc_table_compress_2 wurden den Tablespaces zugeordnet.

```
CREATE TABLESPACE hc_compress1 DATAFILE '...\hc_compress1.dbf' SIZE 1G
REUSE AUTOEXTEND ON NEXT 1G MAXSIZE UNLIMITED DEFAULT COMPRESS FOR ALL
OPERATIONS;
CREATE TABLESPACE hc_compress2 DATAFILE '...\hc_compress2.dbf' SIZE 1G
REUSE AUTOEXTEND ON NEXT 1G MAXSIZE UNLIMITED DEFAULT COMPRESS FOR
DIRECT_LOAD OPERATIONS;
```

Jetzt konnten wir mit CTSA (create Table as select) die Daten in die vorbereiteten Tablespaces laden und dabei automatisch komprimieren.

Komprimierung Teil 1

Im ersten Schritt wurden die ~ 200 mio Datensätze mit CTAS in die mit COMPRESS FOR ALL OPERATIONS komprimierte Tabelle hc_table_compress_1 geladen:

```
CREATE TABLE testhc.hc_table_compress_1 TABLESPACE hc_compress1 as select *  
from testhc.hc_table_normal;
```

Im Anschluß in die mit COMPRESS FOR DIRECT_LOAD OPERATIONS komprimierte Tabelle hc_table_compress_2.

```
CREATE TABLE testhc.hc_table_compress_2 TABLESPACE hc_compress2 as select *  
from testhc.hc_table_normal;
```

Da für Siemens nur der Lese und Verarbeitungsprozesse kritisch sind, waren die Insertzeiten für Siemens nicht relevant, das System war jecoch in der Lage mit ca. 0,8 GB/sec auf die Platten zuzugreifen und so konnten die 40 GB in ca. 1 Minute geschrieben werden. Jetzt war die spannende Frage wie viel Platzbedarf für diese beiden Komprimierungsarten notwendig ist.

ASCII on disk	30.100 MB
UNCOMPRESSED	40.067 MB (Referenzwert)
COMPRESS FOR ALL OPERATIONS	29.953 MB
COMPRESS FOR DIRECT_LOAD OPERATIONS	26.817 MB

Zu COMPRESS FOR ALL OPERATIONS schreibt Oracle im Oracle® Database New Features Guide 11g Release 1 (11.1)

(Quelle: http://download.oracle.com/docs/cd/B28359_01/server.111/b28279/chapter1.htm#FEATURENO07585)

The Oracle compression feature allows heap tables to be stored in a compressed format, resulting in significant savings in disk storage, I/O, and redo logs. This feature extends the compression feature to OLTP environments so that compression works seamlessly for all DMLs.

Oracle's heap segment compression (HSC) has been well received in data warehousing environments. With this feature compression is extended to OLTP environments so that compression works well with all DMLs including conventional INSERT, UPDATE, and DELETE.

Jetzt steht also neben dem bereits bekannten Kompressionsverfahren COMPRESS FOR DIRECT_LOAD OPERATIONS auch ein Verfahren für DML Umgebungen (OLTP) zur Verfügung. Auch wenn beide Verfahren technisch sehr unterschiedlich arbeitende, liefern sie doch im vorliegenden Siemens Fall vergleichbare Ergebnisse von 25% Datenreduktion für COMPRESS FOR ALL OPERATIONS und 33% Datenreduktion bei COMPRESS FOR DIRECT_LOAD OPERATIONS.

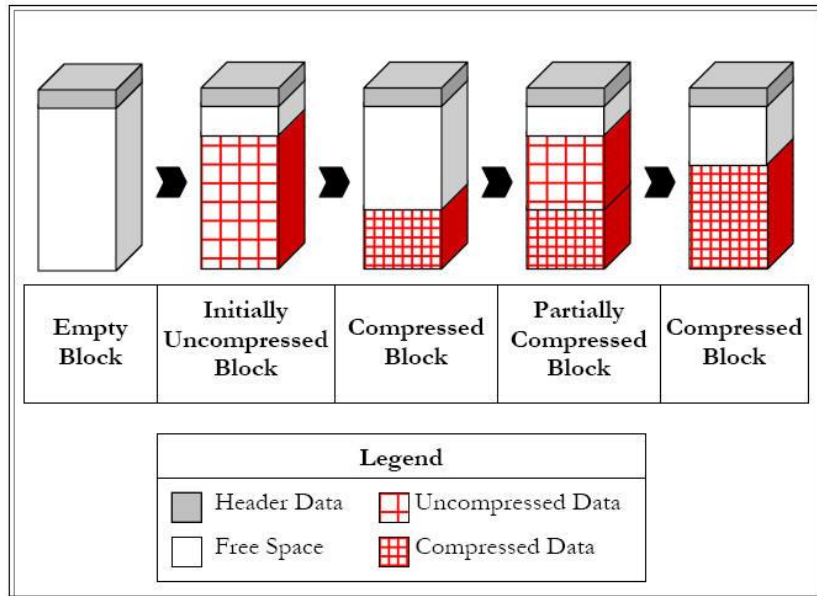


Abb. 2: Oracle's heap segment compression (HSC)

Neben der reinen Platzersparnis war jedoch auch wichtig für Siemens zu erfahren wie sich CPU Last, Speicherlast, Performance und Kommunikationsoverhead verhalten. Dazu haben wir ein identisches select Statement bei allen drei Tabellen (hc_table_normal, hc_table_compress_1 und hc_table_compress_2) durchgeführt und die Werte mithilfe von Automatic Workload Repository (AWR) und Microsoft Management Console (MMC) gemessen.

```
set echo on
alter system flush shared_pool;
alter system flush buffer_cache;
alter system switch logfile;
exec dbms_workload_repository.create_snapshot();
set timing on
select count(component1) from testhc.hc_table_normal;
set timing off
exec dbms_workload_repository.create_snapshot();
```

```
screenshot Performancemonitor
...\report01.rtf
```

```
@...\product\11.2.0\dbhome_1\RDBMS\ADMIN\awrrpt.sql "...report01.html"
```

Neben der erwarteten Platzersparnis konnten wir so auch eine vergleichbare Abfragezeitverkürzung messen.

UNCOMPRESSED	Abfragezeit: 00:01:06.75 (Referenzwert)
COMPRESS FOR ALL OPERATIONS	Abfragezeit: 00:00:54.51 (- 18%)
COMPRESS FOR DIRECT_LOAD OPERATIONS	Abfragezeit: 00:00:43.93 (-33%)

Memory- und CPUverbrauch waren unverändert, wenn man von der kürzeren Statementlaufzeit absieht. Abschließend lohnt sich noch ein Blick auf die drei unterschiedlichen Top 5 Timed Foreground Events aus den Automatic Workload Repository (AWR) Reports.

Top 5 Timed Foreground Events

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
direct path read	37,836	47	1	47.05	User I/O
DB CPU		26		25.50	
db file sequential read	2,887	19	7	19.28	User I/O
db file scattered read	448	3	6	2.62	User I/O
control file sequential read	222	2	9	2.03	System I/O

Abb. 3: UNCOMPRESSED Top 5 Timed Foreground Events

Top 5 Timed Foreground Events

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
DB CPU		37		42.00	
direct path read	17,877	23	1	26.83	User I/O
db file sequential read	2,910	17	6	19.17	User I/O
db file scattered read	461	3	6	3.37	User I/O
control file sequential read	154	0	3	0.53	System I/O

Abb. 3: COMPRESS FOR ALL OPERATIONS Top 5 Timed Foreground Events

Top 5 Timed Foreground Events

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
DB CPU		33		47.65	
db file sequential read	2,006	15	7	21.30	User I/O
direct path read	11,332	14	1	19.43	User I/O
db file scattered read	326	2	5	2.32	User I/O
log file sync	11	1	58	0.91	Commit

Abb. 3: COMPRESS FOR DIRECT_LOAD OPERATIONS Top 5 Timed Foreground Events

Zwei Dinge fallen sofort auf: Bei UNCOMPRESSED ist der Zeitkritische Prozess "User I/O". Bei COMPRESS FOR ALL OPERATIONS und FOR_LOAD OPERATIONS nur noch die CPU. Die ca. 38tsd. direct path reads belegen ca. 47% der Zeit bei UNCOMPRESSED. Bei COMPRESS FOR ALL OPERATIONS bzw. FOR DIRECT_LOAD OPERATIONS sind es nur noch ca. 18tsd. bzw. 11tsd. und ca. 27% bzw. 19% der Gesamtzeit. Der Flaschenhals ist nun also nicht mehr der Datenzugriff sondern die Datenweiterverarbeitung, obwohl diese nominal auch nicht ansteigt sondern ebenfalls fällt. Damit lässt sich der gesamte Prozess beschleunigen und bietet die Möglichkeit diesen durch Einsatz von mehr CPU Leistung noch stärker zu optimieren.

Komprimierung Teil 2

Oracle bietet jedoch noch weitere Möglichkeiten an die Daten zu komprimieren.

High compression level for query operations
Low compression level for query operations
High compression level for archive operations
Low compression level for archive operations

Diese stehen jedoch nur auf Oracle Exadata zur Verfügung. Siemens hatte zwar keine Oracle Exadata jedoch starkes Interesse an den damit erreichbaren Kompressionsraten. Wir konnten hier zumindest mit einem gutes estimate weiterhelfen. Dazu legten wir den Tablespace SCRATCHTBS an.

```
CREATE TABLESPACE SCRATCHTBS DATAFILE '...\SCRATCHTBS.dbf' SIZE 10G REUSE
AUTOEXTEND ON NEXT 1G MAXSIZE UNLIMITED DEFAULT NOCOMPRESS;
```

Mit dem folgenden Statement

```
=====
== COMP_FOR_QUERY_HIGH
=====
set serveroutput on

declare
  b_cmp          PLS_INTEGER;
  b_uncmp        PLS_INTEGER;
  row_cmp        PLS_INTEGER;
  row_uncmp      PLS_INTEGER;
  cmp_ratio      NUMBER;
  cmp_str        VARCHAR2(200);
begin
  DBMS_COMPRESSION.GET_COMPRESSION_RATIO (
  SCRATCHTBSNAME      => 'SCRATCHTBS',
  OWNNAME             => 'TESTHC',
  TABNAME             => 'HC_TABLE_NORMAL',
  PARTNAME            => '',
  COMPTYPE            => DBMS_COMPRESSION.COMP_FOR_QUERY_HIGH,
  BLKCNT_CMP          => b_cmp,
  BLKCNT_UNCMP        => b_uncmp,
  ROW_CMP             => row_cmp,
  ROW_UNCMP           => row_uncmp,
  CMP_RATIO           => cmp_ratio,
  COMPTYPE_STR        => cmp_str);

  dbms_output.put_line('# Blocks compressed      => '|| b_cmp);
  dbms_output.put_line('# Blocks uncompressed => '|| b_uncmp);
  dbms_output.put_line('Ratio              => '|| cmp_ratio);
  dbms_output.put_line('Komprimierungstyp  => '|| cmp_str);
end;
/
```

wurden die unkomprimierten Testdaten der Tabelle HC_TABLE_NORMAL innerhalb des Tablespaces SCRATCHTBS mit den oben angegebenen vier Kompressionsarten komprimiert. Im Handbuch ([Oracle® Database PL/SQL Packages and Types Reference 11g Release 2](#)) steht dazu:

The *DBMS_COMPRESSION* package gathers compression-related information within a database environment. This includes tools for estimating compressibility of a table for both partitioned and non-partitioned tables, and gathering row-level compression information on previously compressed tables. This gives the user with adequate information to make compression-related decision.

Wir konnten die nachfolgenden Werte ermitteln:

ASCII on disk	30.100 MB
UNCOMPRESSED	40.067 MB (Referenzwert)
COMPRESS FOR ALL OPERATIONS	29.953 MB / 25% Datenreduktion
COMPRESS FOR DIRECT_LOAD OPERATIONS	26.817 MB / 33% Datenreduktion
High compression level for query operations	3.225 MB / 92% Datenreduktion
Low compression level for query operations	7.369 MB / 82% Datenreduktion
High compression level for archive operations	1.659 MB / 96% Datenreduktion
Low compression level for archive operations	2.618 MB / 93% Datenreduktion

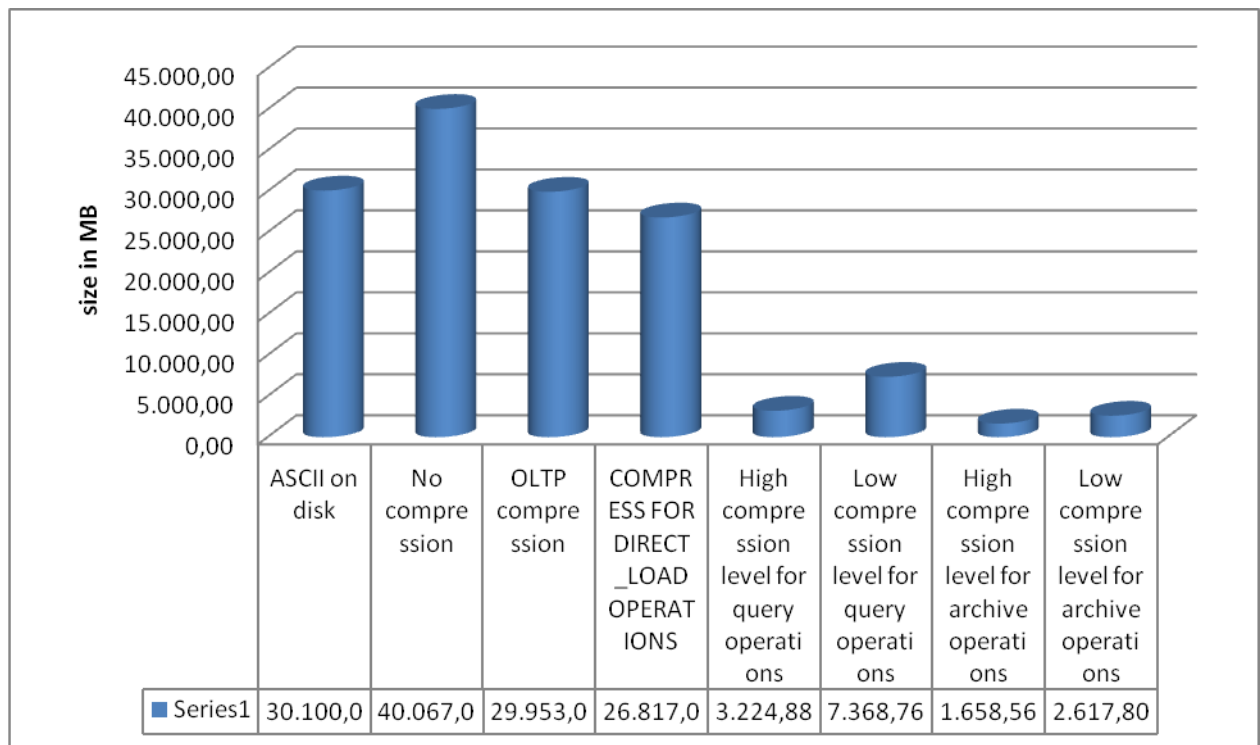


Abb. 4: Kompressionswerte im Überblick

Ausblick

Siemens ist sicher, mit den Oracle Kompressionsmöglichkeiten das richtige Instrument für einen wirtschaftlichen Betrieb der Remote Service Platform gefunden zu haben. In einem weiteren Proof of Concept (PoC) werden diese Werte auf einer Oracle Exadata validiert.

Die Seitenzahl wird von uns eingefügt!

Kontaktadresse:

Heinz Mielimonka
ORACLE Deutschland B.V. & Co. KG
Lina-Ammon-Str.19
D- 90471 Nürnberg

Telefon: +49 (0) 911 98182 152
Fax: +49 (0) 911 98182 111
E-Mail heinz.mielimonka@oracle.com
Internet: www.oracle.com/de