ORACLE

ZFS STORAGE APPLIANCE

An Oracle Technical White Paper

May 2014

Implementing Hybrid Columnar Compression on the Oracle ZFS Storage Appliance

Synopsis: Hybrid Columnar Compression is a unique and effective compression option for Oracle Databases that reside on Oracle Storage. This solution brief provides technical guidance to effectively implement Hybrid Columnar Compression on the Oracle ZFS Storage Appliance.

ORACLE

Introduction

This paper illustrates the implementation of Hybrid Columnar Compression (HCC) for Oracle Database tables and partitions which reside on an Oracle ZFS Storage Appliance. It covers implementation of Hybrid Columnar Compression on Oracle ZFS Storage Appliance-resident direct Network File System (dNFS) shares, Automatic Storage Management (ASM) disk groups on dNFS shares, and ASM disk groups on LUNs (iSCSI or Fibre Channel [FC]).

Highlighted in this paper are:

- Overview of the Oracle ZFS Storage Appliance
- Overview of Hybrid Columnar Compression
- Implementation of Hybrid Columnar Compression on dNFS shares without ASM
- Implementation of Hybrid Columnar Compression using ASM on dNFS shares
- Implementation of Hybrid Columnar Compression using ASM on iSCSI and Fibre Channel LUNs

About the Oracle ZFS Storage Appliance

The Oracle ZFS Storage Appliance is an enterprise multiprotocol storage platform that offers extensive functionality. With cost-effective, yet powerful controllers, enterprise serial-attached SCSI (SAS) storage hardware components, and an advanced operating system employing Hybrid Storage Pools, the Oracle ZFS Storage Appliance delivers high performance at a lower price point than major competitors. The Oracle ZFS Storage Appliance also includes a rich set of data services that can help speed up implementation and simplify data management.

Hybrid Storage Pool technology powers the superior performance of the Oracle ZFS Storage Appliance. Using an intelligent and adaptive set of algorithms to manage read operations, the Oracle ZFS Storage Appliance is able to cache data stored on disk in either direct random access memory (DRAM) or flash-based memory for low latency, high I/O, and high throughput access. Similarly, write operations can be made to low latency, nonvolatile flash so that they can be quickly acknowledged, allowing the system to move on to the next operation more quickly. As the data movement occurs, end-to-end Oracle ZFS Storage Appliance checksumming prevents silent data corruption. This technology allows the Oracle ZFS Storage Appliance to extract maximum performance from the hardware, enabling extreme performance while minimizing price.

The rich set of data services and appliance options further enhance the economic benefits of the Oracle ZFS Storage Appliance. All protocols (such as Common Internet File System (CIFS), Internet Small Computer System Interface (iSCSI), InfiniBand (IB), and Fibre Channel (FC) and others) are included in the base price of the system. Various data protection options and several data compression options are included in the base price. Customers can choose between high-capacity and high-performance SAS disk drives and various memory (DRAM and flash) options to optimize for their performance and capacity needs while staying within budget. Advanced data services like remote replication and snapshot cloning are also available.

Oracle offers several models of the Oracle ZFS Storage Appliance to meet the scalability and availability needs of today's demanding applications.

Current hardware models include the Oracle ZFS Storage ZS3-2 (for midrange enterprise storage workloads) and the Oracle ZFS Storage ZS3-4 (for high-end enterprise storage workloads).
For specific details on the current Oracle ZFS Storage systems' hardware and configuration options, see the Oracle storage product documentation pages.

The new Oracle ZFS Storage platforms offer faster CPUs, bigger flash cache, larger storage capacity, and better throughput to meet the storage requirements of mission-critical applications.

About Hybrid Columnar Compression

Hybrid Columnar Compression is an Oracle Database compression technology which enables the highest levels of data compression and provides enterprises with cost savings and performance improvements due to reduced I/O. Average storage savings can range from 6x to 15x, depending on which Hybrid Columnar Compression level is implemented. Hybrid Columnar Compression gives database administrators the flexibility to select from four different levels of Data Warehouse or Archive compression to balance storage savings and I/O reduction with CPU resource utilization. Hybrid Columnar Compression can be enabled on Oracle Databases residing on dNFS shares without ASM on an Oracle ZFS Storage Appliance beginning with Oracle Database Enterprise Edition 11.2.0.3.

HCC can be enabled for Oracle Databases residing in ASM on iSCSI or Fibre Channel LUNs, or dNFS shares, on a ZFS Storage Appliance beginning with Oracle Database 11*g*, Enterprise Edition 11.2.0.4. There is no extra license necessary for Hybrid Columnar Compression. Oracle Database Enterprise Edition is required.

Hybrid Columnar Compression uses a unique physical layout of data stored in database blocks to enhance the compressibility of that data. Traditionally, data has been organized within a database block in a 'row' format where all column data for a particular row is stored sequentially within a database block. Another approach stores data in a 'columnar' format, in which data is organized and stored by column. Hybrid Columnar Compression uses a unique combination of both row and columnar methods for storing data, which optimizes compression while minimizing the performance shortfalls of a pure columnar format. A logical construct called the compression unit (Figure 1) is used to store a set of hybrid columnar-compressed rows. When data is loaded, column values for a set of rows are grouped together and compressed. After the column data for a set of rows has been compressed, it is stored in a compression unit.

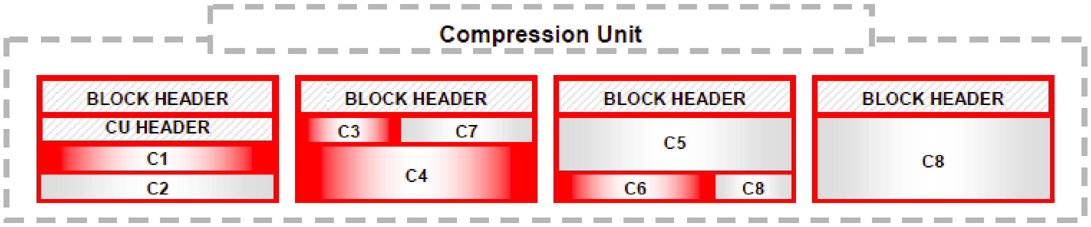

Figure 1. Example of compression unit

Hybrid Columnar Compression Algorithms

There are four different possible Hybrid Columnar Compression algorithms that can be applied to a table or partition. Two are Data Warehouse compression algorithms, suitable for data warehouse access profiles, and two are Archive algorithms more suited to relatively inactive tables or partitions:

- Data Warehouse:
  - Query Low
  - Query High
- Archive:
  - Archive Low
  - Archive High

Figure 2 provides visual comparisons.

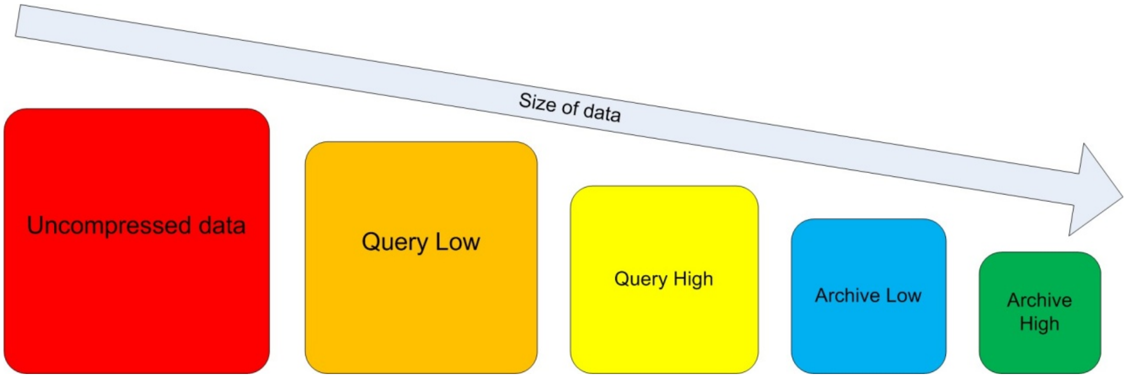

Figure 2. Relative size of data for each compression algorithm

Data is compressed using the database server CPU when loaded using bulk load operations. Examples of commonly used bulk load operations are:

- Insert statements with the APPEND hint
- Parallel DML (parallel INSERT, UPDATE, MERGE, and DELETE)
- Direct Path SQL*LDR (SQL Loader)
- Create Table as Select (CTAS)

Indexes and LOB (Large OBject) segments are not compressed with Hybrid Columnar Compression. Data inserted or updated using any other method is not compressed at the time of the insert. Active tables with frequent updates or inserts could experience rapid growth, as the newly inserted or changed data is not compressed at the time of update or insert.

Hybrid Columnar Compression is normally implemented by creating a table or partition using the "COMPRESS FOR" phrase with one of the Hybrid Columnar Compression options, then loading the table or partition. For example:

```
CREATE TABLE OI_HIGH TABLESPACE COMP COMPRESS FOR QUERY HIGH AS SELECT * FROM
OE.OI;
```

This statement creates a table named `OI_HIGH` in a tablespace named `COMP`, sets the table compression attribute to `QUERY HIGH`, and loads the table with the rows from table `OE.OI`. The result is a table with data compressed using the `HCC QUERY HIGH` algorithm.

An existing table can be compressed in place using the `ALTER TABLE…MOVE COMPRESS FOR` command. For example:

```
ALTER TABLE OI_TABLE MOVE COMPRESS FOR ARCHIVE LOW;
```

Any indexes over a table compressed using `ALTER TABLE…MOVE COMPRESS FOR` will require that the indexes be rebuilt after the `MOVE` operation completes.   To determine which indexes need to be rebuilt, it is recommended to issue a command before the `ALTER TABLE` is issued to display any indexes that are already unusable, then issue the `ALTER TABLE…MOVE` operation, then display the unusable indexes again to determine which indexes were invalidated by the operation.

Before implementing Hybrid Columnar Compression, it is recommended to run the Oracle Advanced Compression Advisor tool (DBMS_COMPRESSION), which is a PL/SQL package included with Oracle Database 11*g* Release 2 that is used to estimate potential storage savings for Hybrid Columnar Compression based on analysis of a sample of data. This tool compresses a subset of the target data using the actual compression algorithms used by Hybrid Columnar Compression to judge the effectiveness of each option. It provides a good estimate of the actual results that may be obtained after implementing Hybrid Columnar Compression. The Advanced Compression Advisor requires a minimum of one million records in a table/partition in order to estimate Hybrid Columnar Compression ratios.

Hybrid Columnar Compression Implementation Options

The three options for implementing Hybrid Columnar Compression on the Oracle ZFS Storage Appliance are:

- Hybrid Columnar Compression on dNFS shares without ASM
- Hybrid Columnar Compression using ASM on dNFS shares
- Hybrid Columnar Compression using ASM on iSCSI and Fibre Channel LUNs

Hybrid Columnar Compression on dNFS Shares

Oracle Direct NFS (dNFS) is a customized NFS client embedded in the database kernel beginning with Oracle Database 11*g* Release 1. It operates on database shares supporting NFS V3 protocol that have been designated for dNFS access. Oracle Database segments at version 11.2.0.3 or higher, residing on dNFS shares on an Oracle ZFS Storage Appliance, are eligible for Hybrid Columnar Compression when ASM is not used. With Oracle Database release 11.2.0.4, Hybrid Columnar Compression can be used on ASM-managed database segments that reside on dNFS shares.

The steps involved in using Hybrid Columnar Compression on dNFS shares on an Oracle Database Appliance are:

1. Enable dNFS on all database homes, and ASM homes if ASM is used.
2. Create and mount shares.
3. Create an `oranfstab` file.
4. Enable and configure SNMP on each Oracle ZFS Appliance where Hybrid Columnar Compression-enabled dNFS shares will reside.

5. If using ASM, create an ASM disk group containing dNFS-enabled shares, and configure the disk group to be enabled for Hybrid Columnar Compression. Create database tablespaces on Oracle ZFS Storage dNFS shares or disk groups and create tables or partitions using the COMPRESS FOR options that enable Hybrid Columnar Compression.

Enable dNFS on database hosts

On version 11.2.0.3 and later Oracle Databases, you can use a `make` command to enable dNFS.

If ASM is going to be used on dNFS shares, dNFS must be enabled in both the `ASM GRID_ HOME` and the database `ORACLE_HOME`. In an Oracle RAC environment, dNFS must be enabled on all hosts in the RAC. To enable dNFS, issue this command:

```
make –f $ORACLE_HOME(or $GRID_HOME)/rdbms/lib/ins_rdbms.mk dnfs_on
```

To disable dNFS, issue this command:

```
make –f $ORACLE_HOME(or $GRID_HOME)/rdbms/lib/ins_rdbms.mk dnfs_off
```

Create and mount shares

Refer to the Oracle Technology Network web site My Oracle Support (MOS) (see References at the end of this document) and note 359515.1 for the latest information on mount options for NFS shares, and 1620238.1 for the latest information on using files on NFS shares as ASM devices.

Use the following steps to create and mount shares:

1. Create a share on the Oracle ZFS Storage Appliance. For this example, a share suitable for data file storage is created, since Hybrid Columnar Compression applies to table data. The share should have a ZFS Database record size of 8 K, Synchronous write bias of throughput, and Cache device usage of "All data and metadata."
2. Create a directory on the database host and an entry in the system `/etc/fstab` with the following mount options: `rw,bg,hard,nointr,rsize=1048576,wsize=1048576,tcp,actimeo=0, nfsvers=3,timeo=600`

Example:

```
aie-7320d-h1:/export/hccasm /mnt/hccasm nfs
rw,bg,hard,nointr,tcp,nfsvers=3,timeo=600,rsize=1048576,wsize=1048576,
actimeo=0
```

This example is for a single-instance database on Linux. Refer to MOS for specific options for other operating systems.

3. Mount the share using the `mount` command.

4. Change the ownership of the share to the oracle user and group; for example, `oracle:oinstall`.

Example:

```
chown –R oracle:oinstall /mnt/hccshr
chmod –R 660 /mnt/hccshr
```

Create an `oranfstab` file

There are several options for specifying the dNFS mountpoint information, but the recommendation is to use a file called `oranfstab`. This file can reside in two locations: in the `/etc` directory, or in the `$ORACLE_HOME/dbs`. If it resides in both locations, the version of `oranfstab` in `$ORACLE_HOME/dbs` will be used. If there is no `oranfstab` to be found, dNFS will still be enabled on mountpoints found in the system `mtab` which contain database segments, but you will not have the option of specifying multiple ports or paths for the access. If ASM is to be used with dNFS shares, it is recommended that `oranfstab` reside only in the `/etc` directory so that the ASM instance and the database instance have a consistent copy of `oranfstab`.

In the `oranfstab`, the following attributes can be specified:

- `server` – The Oracle ZFS Storage Appliance name.
- `path` – Up to four network paths to the Oracle ZFS Storage Appliance, specified either by IP address or by DNS name.
- `local` – Up to four local paths on the database host, specified by IP address or by name, as displayed using the `ifconfig` command run on the database host.
- `export` – The exported path from the Oracle ZFS Storage Appliance.
- `mount` – The corresponding local mount point for the exported volume.
- `dontroute` – An optional parameter specifying that outgoing messages should not be routed by the operating system, but sent using the IP address they are bound to. (Please note that this attribute does not work on Linux with multiple paths in the same subnet.)
- `mnt_timeout`: An optional parameter specifying (in seconds) the time for which the Direct NFS Client should wait for a successful mount before timing out. The default is 600 seconds.

Example 1:

```
server: myzfsstorageappliance
local: 10.10.45.12
path: 10.10.35.12
local: 10.10.45.13
path: 10.10.35.13
export: /export/hccshare mount: /mnt/hccshare
export: /export/hccshare1 mount: /mnt/hccshare1
```

Example 2:

```
server: myzfsstorageappliance
path: 10.10.35.12
path: 10.10.35.13
export: /export/hccshare mount: /mnt/hccshare
export: /export/hccshare1 mount: /mnt/hccshare1
```

Enable and configure SNMP on the Oracle ZFS Storage Appliance

SNMP is used to verify that the dNFS shares to be used for Hybrid Columnar Compression reside on an Oracle ZFS Storage Appliance.

1. Verify that the SNMP service is active on the Oracle ZFS Storage Appliance:
   - Log in as root to the Oracle ZFS Storage Appliance BUI.
     - Click Configuration->Services.
     - In the System Settings section, verify that SNMP is online.
   - Click on the SNMP link in the Services section to access the SNMP Services screen.
     - Set the SNMP community name to public.
     - Set the authorized network to 0.0.0.0/0.
     - Set the appliance contact to a text string.
     - Add a trap destination of 127.0.0.1.
     - Click "Apply."
2. From each database host, verify that the Oracle ZFS Storage Appliance identifying message can be received by issuing the `snmpget` command:

   ```
   snmpget -v1 -c public <ZFS Appliance DNS or IP> 1.3.6.1.4.1.42.2.225.1.4.2.0
   ```

   Example:

   ```
   [oracle@server ~]$ snmpget -v1 -c public zba-h1 1.3.6.1.4.1.42.2.225.1.4.2.0

           SNMPv2-SMI::enterprises.42.2.225.1.4.2.0 = STRING: "Sun ZFS Storage 7420"
   ```

Hybrid Columnar Compression Using ASM on dNFS Shares

Hybrid Columnar Compression can be implemented on database segments residing in ASM disk groups that are on dNFS shares, beginning with Grid Infrastructure and Oracle Database 11*g*, release 11.2.0.4. Refer to the Oracle Grid Infrastructure Installation Guide for more information.

If the share is going to be used as an ASM disk resource, perform all of the steps above to enable dNFS on the share, then perform the following additional steps:

1. Create a file in a subdirectory on the NFS share to be used as the ASM device. The file must be zeroed out using the `dd` command. This example creates a subdirectory called `nfsdg` and a 1 gigabyte file called `disk1` in subdirectory `nfsdg` on the share mounted on `/mnt/hccasm`:

   ```
   mkdir /mnt/hccasm/nfsdg
   dd if=/dev/zero of=/mnt/hccasm/nfsdg/disk1 bs=1024k count=1000 oflag=direct
   ```

2. Change permissions and ownership of the directory and the file to be the Oracle Grid Infrastructure owner and group:

   ```
   chown –R oracle:oinstall /mnt/hccasm/nfsdg
    chmod –R 660 /mnt/hccasm/nfsdg
   ```

3. Edit the ASM disk discovery string to specify a regular expression that matches the file name you created, in this example:

   ```
   /mnt/hccasm/nfsdg/*
   ```

4. Use ASMCA, SQL commands, or `oracleasm` commands to create an ASM disk group using the discovered file as an external redundancy volume. After creating the disk group, set the compatibility attributes and the `storage.type` attribute.

   ```
   create diskgroup NFS1 external redundancy disk '/mnt/hccasm/nfsdg/disk1';
   ```

5. Issue commands to set attributes on the ASM disk group:

   ```
           > sqlplus / as sysasm

           SQL*Plus: Release 11.2.0.4.0 Production on Fri Mar 21 16:03:43 2014

           Copyright (c) 1982, 2013, Oracle.  All rights reserved.


           Connected to:
           Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit
   Production
           With the Automatic Storage Management option

           SQL> create diskgroup NFS1 external redundancy disk
   '/mnt/hccasm/nfsdg/disk1';

           Diskgroup created.
   ```

```
SQL> alter diskgroup NFS1 set attribute 'compatible.asm'='11.2.0.4';

Diskgroup altered.

SQL> alter diskgroup NFS1 set attribute 'compatible.rdbms'='11.2.0.4';

Diskgroup altered.

SQL> alter diskgroup NFS1 set attribute 'storage.type'='ZFSSA';

Diskgroup altered.
```

Verifying the Use of dNFS

Verify that dNFS is being used to access the database you are using. Log in to the database sys user or the ASM sysasm user and issue select commands from the dNFS related system tables:

- `v$dnfs_servers` – Servers accessed using Direct NFS
- `v$dnfs_files` – Files open with dNFS
- `v$dnfs_channels` – Channels (paths) to Oracle ZFS Storage Appliances serving dNFS files
- `v$dnfs_stats` – Performance statistics for dNFS

```
SQL> select * from v$dnfs_servers;

ID
----------
SVRNAME
--------------------------------------------------------------------------------
DIRNAME
--------------------------------------------------------------------------------
   MNTPORT    NFSPORT      WTMAX      RTMAX
---------- ---------- ---------- ----------
         1
aie-7320d-h1
/export/blazemirror0
     62564       2049    1048576    1048576


SQL> select * from v$dnfs_files;

FILENAME
--------------------------------------------------------------------------------
  FILESIZE       PNUM     SVR_ID
---------- ---------- ----------
```

```
/mnt/blazemirror0/hcc/control01.ctl
  10043392         12          1

/mnt/blazemirror0/oradata/fast_recovery_area/hcc/control02.ctl
  10043392         12          1

/mnt/blazemirror0/hcc/system01.dbf
 775954432         10          1
.
.
.
```

Hybrid Columnar Compression Using ASM with iSCSI or Fibre Channel LUNs

Creating LUNs and making them accessible to the operating system is platform dependent and beyond the scope of this document. Refer to “Using Oracle ZFS Storage Appliance iSCSI LUNs in an Oracle Linux Environment” or “Using Oracle ZFS Storage Appliance iSCSI LUNs in an Oracle Solaris Environment” on Oracle Technology Network's NAS Storage Documentation site listed at the end of this document.

IMPORTANT: If using iSCSI with Hybrid Columnar Compression, the fix for bug 17608538 is required.

Once the LUNs are created and recognized by the operating system, add one primary partition using the entire volume other than the area needed for the partition table. Do not create a file system, as ASM will use its own methods to write to the partition.

Proceed to add the LUNs to an ASM disk group:

1. Change permissions and ownership of the `/dev` entries to be the Oracle Grid Infrastructure owner and group (`oracle` and `oinstall` in this example):

   ```
    chown –R oracle:oinstall /dev/sde1
   chmod –R 660 /dev/sde1
   ```

2. Edit the ASM disk discovery string to specify a regular expression that matches the `/dev` entry for the partition/LUN. Example:

   ```
   /dev/sde*
   ```

3. Use ASMCA, SQL commands, or `oracleasm` commands to create an ASM disk group using the discovered LUN as an external redundancy volume. After creating the disk group, set the compatibility attributes and the `storage.type` attribute.

   ```
         > sqlplus / as sysasm
   ```

```
       SQL*Plus: Release 11.2.0.4.0 Production on Fri Mar 21 16:03:43 2014

       Copyright (c) 1982, 2013, Oracle.  All rights reserved.


       Connected to:
       Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit
Production
       With the Automatic Storage Management option

       SQL> create diskgroup LUN1 external redundancy disk '/dev/sde1';

       Diskgroup created.

       SQL> alter diskgroup LUN1 set attribute 'compatible.asm'='11.2.0.4';

       Diskgroup altered.

       SQL> alter diskgroup LUN1 set attribute 'compatible.rdbms'='11.2.0.4';

       Diskgroup altered.

       SQL> alter diskgroup LUN1 set attribute 'storage.type'='ZFSSA';

       Diskgroup altered.
```

Creating a Database Tablespace

Create a database tablespace on Oracle ZFS Appliance dNFS shares or ASM disk groups and create tables or partitions using the COMPRESS FOR options that enable Hybrid Columnar Compression.

Once the shares or LUNs are created, and ASM set up if necessary, tablespaces must be created to contain the tables and/or partitions which will be compressed using Hybrid Columnar Compression. You can specify compression at a tablespace level, or for an individual table or partition. If specified at the tablespace level, then all tables created in that tablespace are compressed by default.

Tablespace on dNFS share example:

```
create bigfile tablespace hccts datafile '/mnt/hccshare/hccts.dbf' size 100M
default compress for query high;
```

Tablespace on ASM example:

```
create bigfile tablespace hccts datafile '+NFS1/hccts.dbf' size 100M default
compress for query high;
```

The more common case is compression of existing individual tables. In the following example, these steps/operations are performed:

1. Determine whether there are any indexes currently in UNUSABLE state. Do this in order to know which indexes will be invalidated by the `ALTER TABLE` command about to be issued, as opposed to those that are already unusable for another reason.
2. Issue an `ALTER TABLE…MOVE COMPRESS FOR` command to compress the table. This could potentially be a long-running operation, and the table is not available for use while the compression is taking place.
3. Determine which indexes have been made unusable by the compression operation.
4. Rebuild indexes if necessary.

Example:

```
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> select index_name from dba_indexes where status = 'UNUSABLE';
```

(This command shows indexes that are already unavailable.)

```
       no rows selected
```

(In this example there are no unavailable indexes prior to compression.)

```
SQL> alter table soe.order_items move compress for query high;
```

(This command applies QUERY HIGH compression to the table.)

```
Table altered.
```

(This is the only indication that compression was successful.)

```
SQL> select index_name from dba_indexes where status = 'UNUSABLE';
```

(This command shows indexes that are unusable after compression.)

```
INDEX_NAME
------------------------------
ORDER_ITEMS_PK
ITEM_ORDER_IX
ITEM_PRODUCT_IX
```

(Three indexes were made unusable by the `ALTER TABLE MOVE` command.)

```
SQL> alter index SOE.ORDER_ITEMS_PK rebuild;
```

(Three indexes should be rebuilt.)

```
Index altered.

SQL> alter index SOE.ITEM_ORDER_IX rebuild;

Index altered.

SQL> alter index SOE.ITEM_PRODUCT_IX rebuild;

Index altered.

SQL> select index_name from dba_indexes where status = 'UNUSABLE';
```

(This command shows indexes which were unusable are now usable.)

```
no rows selected

SQL> select table_name, compression, compress_for from dba_tables where compression = 'ENABLED';
```

(This command shows compression status for a table.)

```
TABLE_NAME                     COMPRESS COMPRESS_FOR
------------------------------ -------- ------------
ORDER_ITEMS                    ENABLED  QUERY HIGH
```

A table may be copied into another and the destination table compressed at the time of copy by using the `CREATE TABLE …COMPRESS FOR…SELECT FROM` operation.

Example:

```
      SQL> create table soe.order_items_archive_high nologging compress for archive high
      as select * from soe.order_items;

      Table created.

      SQL> select table_name, compression, compress_for from dba_tables where
      compression = 'ENABLED';

      TABLE_NAME                     COMPRESS COMPRESS_FOR
      ------------------------------ -------- ------------
      ORDER_ITEMS_ARCHIVE_HIGH       ENABLED  ARCHIVE HIGH
```

```
ORDER_ITEMS                    ENABLED  QUERY HIGH

SQL>
```

A table may be partitioned, and selected partitions can be compressed using HCC. Partitioning can be a complex subject, and the user is advised to consult the *Oracle Database VLDB and Partitioning Guide*. Databases may contain compressed partitions residing on the Oracle ZFS Storage Appliance and uncompressed partitions on other storage.

A simple example of partitioning and using Hybrid Columnar Compression to compress the data that is expected to be less active can be illustrated by the Oracle Database sample schema of HR, which contains a table called JOB_HISTORY. A user could move rows for employees whose `END_DATE` was prior to a certain date into a table compressed using the Archive High compression algorithm.

With one SQL statement, a new partitioned table can be created and the data loaded from an existing table, while Hybrid Columnar Compression is applied to one partition but not the other:

```
SQL> create table hr.job_history_partitioned (employee_id, start_date, end_date,
job_id, department_id)  partition by range (end_date) (partition p1 values less
than (TO_DATE('2006-01-01', 'YYYY-MM-DD')) tablespace SOE compress for archive
high, partition p2 values less than (MAXVALUE) ) as select * from hr.job_history;

Table created.


SQL> select table_name, partition_name, compress_for from dba_tab_partitions where
compression = 'ENABLED';

TABLE_NAME                     PARTITION_NAME                 COMPRESS_FOR
------------------------------ ------------------------------ ------------
JOB_HISTORY_PARTITIONED        P1                             ARCHIVE HIGH
.
.
```

Conclusion

Hybrid Columnar Compression is an Oracle-exclusive technology that is easily deployed on databases residing on the Oracle ZFS Storage Appliance. It offers improvements in storage utilization in almost any database, and can often improve performance for large queries.

References

See the following resources for additional information relating to the products covered in this document.

References to Sun ZFS Storage Appliance, Sun ZFS Storage 7000, and ZFS Storage Appliance all refer to the same family of Oracle ZFS Storage Appliance products. Some cited documentation may still carry these legacy naming conventions.

- Oracle ZFS Storage Appliance Documentation Library, including Installation, Analytics, Customer Service, and Administration guides:
  http://www.oracle.com/technetwork/documentation/oracle-unified-ss-193371.html
- The *Oracle ZFS Storage Appliance Administration Guide* is also available through the Oracle ZFS Storage Appliance help context.
  The Help function in Oracle ZFS Storage Appliance can be accessed through the browser user interface.
- Oracle Support Center
  http://www.oracle.com/support

  Patches and updates downloads from My Oracle Support (MOS)
  (search under Oracle ZFS Storage Software Patches)
- Oracle ZFS Storage Appliance Plug-ins
  http://www.oracle.com/technetwork/server-storage/sun-unified-storage/downloads/zfssa-plugins-1489830.html
- Oracle Storage Product Information
  http://www.oracle.com/us/products/storage/overview/index.html
- Oracle ZFS Storage Appliance Technical White Papers and Solution Briefs
  http://www.oracle.com/technetwork/server-storage/sun-unified-storage/documentation/index.html
- Oracle Database Documentation Library, including:
  Oracle Database VLDB and Partitioning Guide, 11g Release 2
  Oracle Grid Infrastructure Installation Guide(s)
  http://www.docs.oracle.com/cd/E11882_01

ORACLE

Implementing Hybrid Columnar Compression
on the Oracle ZFS Storage Appliance
May 2014, Version 1.0
Author: Application Integration Engineering,
Scott Ledbetter
Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com