# IOUG Tips & Best Practices Booklet

A Compilation of Technical Tips from the Independent Oracle Users Group

IOUG independent oracle users group

eleventh edition ■ www.ioug.org

*Sponsored by* ORACLE

# Contents



*Sponsored by*

ORACLE®


IOUG independent oracle users group

**Executive Editor** ■ Andrew Meade
**Managing Editor** ■ Jaclyn Moriarty
**Design** ■ Patrick Williams

**Headquarters**
Independent Oracle Users Group
330 N. Wabash Ave.
Chicago, IL 60611 USA
Phone: +1.312.245.1579
Fax: +1.312.527.6785
email: ioug@ioug.org

# Savvy on Storage

Welcome to the eleventh edition of the *IOUG Tips & Best Practices Booklet*! I am continually impressed by the breadth and depth of knowledge of the IOUG community. With each release of the *IOUG Tips & Best Practices Booklet*, I am amazed with how much there is to learn about Oracle, and how much the IOUG community shares their knowledge. IOUG *SELECT* and the tip booklet are prime examples of what IOUG is all about: Oracle users helping other Oracle users share their technical knowledge and experiences.

This year's *IOUG Tips & Best Practices Booklet* focuses on Storage. The authors are members in the community — many of them COLLABORATE speakers — and they share their best practices for storage. Topics range from simple tips on initial setup and maintenance, to advanced tips on when and where to use SSD, HHD, and even Oracle's Information and Lifecycle Management for storage tiering. I hope that you find these tips helpful, and I look forward to seeing these speakers at COLLABORATE.

A hearty thank you to all of the authors and reviewers for their efforts in providing high-quality content for *SELECT* publications. Inspired to share your perspective? To submit an article or to sign up to become a reviewer, drop us a note at **select@ioug.org**. If you are a new author, then writing a tip is the best place to start!

Andrew Meade
Editor, *IOUG Tips & Best Practices Booklet*

# Best Practices for SSD in Oracle Databases

By Guy Harrison

Spinning magnetic disks have been a bottleneck in database performance since the earliest digital database systems. The mechanical nature of the spinning magnetic disk has rendered it resistant to "Moore's law," improvements observed in electronic components. Every significant database should have a strategy for leveraging SSD to optimize performance.

## Understand SSD Physics

It's tempting to think of SSDs simply as just "faster hard drives." But the reality is that SSDs have different physical characteristics from spinning magnetic disks. Make sure you understand the difference between MLC and SLC and between PCI and SATA based SSD.

Be aware that when a SSD writes it may need to perform a slow block erase operation. Sophisticated SSD devices use freelists and garbage collection algorithms to reduce the overhead of these writes. However, during sustained write activity, there can be a definite degradation in SSD performance.

## The Economics of SSD

SSDs remain much more expensive than spinning magnetic disks in terms of raw storage. While the price per GB for SSDs is falling continuously, so is the price per GB for magnetic disks. For the foreseeable future, SSDs will offer better economics for *fast data,* but poorer economics for *big data.* For large databases, the best overall economics will be achieved by mixing SSD and magnetic disk.

## The Database Flash Cache

Oracle introduced the Database Flash Cache (DBFC) in Oracle version 11.2, but has made it available only on Oracle operating systems. (Solaris and Oracle Enterprise Linux). Provided you are a using one of these operating systems, the Database Flash cache is probably the easiest way of leveraging SSD in your database configuration.

The Database Flash Cache serves as a secondary cache to the Oracle Buffer Cache. Oracle manages data blocks in the buffer cache using a modified Least Recently Used (LRU) algorithm. Simplistically speaking, blocks age out of the buffer cache if they have not been accessed recently.

The DBFC accelerates single block reads — such as those from index or ROWID lookups — but has no effect on full table scans which bypass the buffer cache.

## Where to Use SSD

Oracle database administrators have a range of options for taking advantage of the performance benefits offered by Solid State Disks:

- As noted earlier, databases on Oracle operating systems can use the Oracle database flash cache. This option is simple to implement and can be effective for read-intensive, index-based workloads.
- The entire database might be deployed on SSD. For smaller databases, this may be the simplest option. However, this option will often be economically impractical for large databases.
- Selected data files, tables, indexes or partitions could be located on Solid State Disk. This requires substantial planning and administration, but is likely to be effective if the correct segments are selected.
- The temporary table space could be relocated to Solid State Disk storage, accelerating the performance of temporary segment I/O, as occurs with disk sort or hashing operations.
- Redo logs are inherently write intensive and have been suggested as candidates for solid state disk storage. The nature of redo log I/O is not a perfect fit for SSD. The sequential write workload of redo log I/O is a perfect fit for spinning magnetic disk, since no seek time is required for each subsequent I/O. However, for SSDs, redo log I/O can tax the garbage collection mechanisms and lead to SSD write degradation.

## Flash and Exadata

Oracle Exadata systems combine SSD and magnetic disk in order to provide a balance between economics of storage and high performance. In an Exadata system, flash SSD is contained in the storage cells only. There is no SSD configured within the compute nodes.

The default configuration for Exadata flash is as in the Exadata Smart Flash Cache. The Exadata Smart Flash Cache is analogous to the Oracle Database Flash Cache, but has a significantly different architecture. The primary purpose of the Exadata Smart Flash Cache is to accelerate read I/O for database files. This is done by configuring flash as a cache over the grid disks that service datafile read I/O. The cache can act both as a read cache and a write-back cache, optimizing both read and write operations.

Exadata also uses flash to reduce redo log "outlier" response times by allowing redo log writes to succeed once the write has completed either to the grid disk or to flash.

### About the Author

**Guy Harrison** is CTO at Southbank Software, a partner at Toba Capital and a software professional with more than 20 years of experience in database design, development, administration and optimization. He is author of *Next Generation Databases, Oracle Performance Survival Guide,* and *MySQL Stored Procedure Programming,* as well as many other books and articles on database technology. Harrison is a MongoDB certified DBA and Developer and an Oracle ACE. This tip was extracted from his chapter on SSD in the *Oracle Database Problem Solving and Troubleshooting Handbook*.

# Utilizing Device Mapper Multipath

By Roger Lopez

The purpose of device mapper multipath (DM-multipath) is to provide the ability to aggregate multiple I/O paths to a newly created device mapper mapping to achieve high availability, I/O load balancing and persistent naming.

## Setting Up DM-Multipath

The following procedures provide the best practices to installing and configuring device mapper multipath devices.

**NOTE:** Ensure Oracle database volumes are accessible via the operating system prior to continuing with the section below.

1. As the root user, install the *device-mapper-multipath* package using the **yum** package manager.

   ```
   # yum install device-mapper-multipath
   ```

2. Run the *mpathconf* command to create a generic multipath.conf file found within */etc/*

   ```
   # mpathconf --enable
   ```

3. Capture the scsi id of the local disk(s) on the system as local disk(s) are not to be included for multipathing. An example is shown below.

   ```
   # /usr/lib/udev/scsi_id --whitelisted --replace-whitespace --device=/dev/sda
   3600508b1001030353434363646301200
   ```

4. Uncomment and modify the blacklist section within the */etc/multipath.conf* file to include the scsi id of the local disk(s) on the system. Once complete, save the changes made to the *multipath.conf* file.

   ```
   blacklist {
         wwid 3600508b1001030353434363646301200
          devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
          devnode "^hd[a-z]"
   }
   ```

5. Start the *multipath* daemon.

```
# systemctl start multipathd.service
```

6. Enable the *multipath* daemon to ensure it is started upon boot time.

```
# systemctl enable multipathd.service
```

7. Identify the *dm- device, size,* and *WWID* of each *device mapper* volume for Oracle data disks and recovery disks. In this example, volume *mpathb* is identified via the following command:

```
# multipath -ll
```


The Multipath Device (mpathb) properly identifies the current multipath alias name, size, WWID and dm device. This information is required for the application of a custom alias to each volume as shown in step 9.

8. Uncomment the defaults section found within the */etc/multipath.conf* file.

```
defaults {
      udev_dir                /dev
       polling_interval       10
       selector               "round-robin 0"
       path_grouping_policy   multibus
       prio                   alua
       path_checker           readsector0
       rr_min_io              100
       max_fds                8192
       rr_weight              priorities
       failback               immediate
       no_path_retry          fail
       user_friendly_names    yes
}
```

**NOTE:** The standard options can be customized to better fit your storage array capabilities. Check with your storage vendor for details.

Uncomment the multipath section found within the */etc/multipath.conf* file and create an alias for each device mapper volume in order to enable persistent naming of those volumes. Once complete, save the changes made to the *multipath.conf* file. The output should resemble the example below. The example below has four volumes used for the Oracle database.

```
multipaths {
        multipath {
                wwid                    3600c0ff000d7e7a899d8515101000000
                alias                   db1
        }
        multipath {
                wwid                    3600c0ff000dabfe5a7d8515101000000
                alias                   db2
        }
        multipath {
                wwid                    3600c0ff000d7e7a8dbd8515101000000
                alias                   fra
        }
        multipath {
                wwid                    3600c0ff000dabfe5f4d8515101000000
                alias                   redo
        }
}
```

9. Restart the *device mapper multipath* daemon.

```
# systemctl restart multipathd.service
```

10. Verify the *device mapper* paths and aliases are displayed properly. Below is an example of one *device mapper* device labeled *fra.*

```
# multipath –ll
fra (3600c0ff000d7e7a89e85ac5101000000) dm-10 HP,MSA2324fc
size=186G features='1 queue_if_no_path' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=130 status=active
| |- 3:0:0:3 sdd  8:48   active ready running
| |- 3:0:1:3 sdh  8:112  active ready running
| |- 4:0:0:3 sdt  65:48  active ready running
| `- 4:0:1:3 sdx  65:112 active ready running
`-+- policy='round-robin 0' prio=10 status=enabled
  |- 3:0:2:3 sdl  8:176  active ready running
  |- 3:0:3:3 sdp  8:240  active ready running
  |- 4:0:2:3 sdab 65:176 active ready running
  `- 4:0:3:3 sdaf 65:240 active ready running
```

## About the Author

**Roger Lopez** is a principal software engineer in Red Hat's Solutions Architecture group. This team identifies high-value solution stacks base on input from sales, marketing and engineering teams and develops reference architectures for internal and external customers. Lopez is a Red Hat Certified Engineer (RHCE) with more than ten years of computer industry experience at Dell and Red Hat in Oracle Database solutions. He has presented at several tech events including Oracle OpenWorld, COLLABORATE and Red Hat Summit.

# Exadata Storage Maintenance

By Andrew Meade

Managing an Exadata Server is a great way to jump from being a normal DBA to a great DMA (Database Machine Administrator), and get into the nitty-gritty details of storage administration. This tip will share some Exadata Storage maintenance jobs, how to manage them and at which logs to look.

I support an I/O intensive Data Warehouse that builds every night. It is fairly consistent, except when impacted by two Exadata Storage maintenance jobs: Exadata Battery Learn Cycle and the Exadata Hard Disk Scrubbing. Note that Exadata Hard Disk scrubbing is different than ASM disk scrubbing.

## Exadata Battery Learn Cycle

The Exadata Battery Learn Cycle runs once per quarter to perform a discharge and charge of the controller battery. During the maintenance, the Flash Cache Mode changes from Write-Back to Write-Through. Write-Back Flash Cache provides the ability to both read and write I/O directly to flash disks. This is safe in case of power loss as the battery backup will allow time for the writes in the Flash Cache to be committed to the Hard Disk. In Write-Through mode, all write I/O is written directly to the Hard Disk, which is significantly slower than writing to Flash Cache first.

### Logs and schedule

The effects of Write-Through mode can be seen in the database alert log errors under high I/O.

```
ORA-27626: Exadata error: 2201 (IO cancelled due to slow/hung disk)
NOTE: ASM has redirected some slow reads to mirror sides to improve performance.
```

You can connect to the cell nodes and use list the alert history command at the CellCLI prompt.

```
CellCLI> list alerthistory
             15_1    2016-10-17T04:00:31-07:00       info    "The HDD disk controller battery is
performing a learn cycle. Battery Serial Number : 1234  Battery Type          : ibbu08  Battery
Temperature   : 27 C  Full Charge Capacity  : 1349 mAh  Relative Charge       : 98%  Ambient
Temperature   : 18 C"
             15_2    2016-10-17T05:13:51-07:00       clear   "All disk drives are in WriteBack
caching mode.  Battery Serial Number : 1234  Battery Type          : ibbu08  Battery Temperature
: 29 C  Full Charge Capacity  : 1348 mAh  Relative Charge       : 71%  Ambient Temperature   : 18
C"
```

By default, the BBU learning cycle is at 2 a.m. on the 17th of every third month (Jan/April/July/Oct). This can be seen at and modified at the CellCLI prompt by running:

```
CellCLI> list cell attributes bbuLearnCycleTime
                         2017-04-17T02:00:00-07:00
CellCLI> alter cell bbuLearnCycleTime='2017-04-17T02:00:00-07:00';
```

## Exadata Disk Scrubbing

A subtler Exadata Maintenance job is the bi-weekly Disk Scrub. This job does not appear in the CellCLI alert history. It only appears in the $CELLTRACE/alert.log.

Disk Scrubbing is designed to periodically validate the integrity of the mirrored ASM extents and thus eliminate latent corruption. The scrubbing is supposed to only run when average I/O utilization is under 25 percent. However, this can still cause spikes in utilization and latency and adversely affect database I/O. Oracle documentation says that a 4TB high capacity hard disk can take 8-12 hours to scrub, but I have seen it run more than 24 hours. Normally, this isn't noticeable as it runs quietly in the background. However, if you have a high I/O workload, the additional 10-15 percent latency is noticeable.

### Logs and schedule

The $CELLTRACE/alert.log on the cell nodes reports the timing and results.

```
Wed Jan 11 16:00:07 2017
Begin scrubbing CellDisk:CD_11_xxxxceladm01.
Begin scrubbing CellDisk:CD_10_xxxxceladm01.
 …
Thu Jan 12 15:12:37 2017
Finished scrubbing CellDisk:CD_10_xxxxceladm01, scrubbed blocks (1MB):3780032, found bad blocks:0
Thu Jan 12 15:42:02 2017
Finished scrubbing CellDisk:CD_11_xxxxceladm01, scrubbed blocks (1MB):3780032, found bad blocks:0
```

You can connect to the cell nodes and alter the Start Time and Interval at CellCLI prompt:

```
CellCLI> alter cell hardDiskScrubStartTime='2017-01-21T08:00:00-08:00';
CellCLI> list cell attributes name,hardDiskScrubInterval
            biweekly
```

## ASM Disk Scrubbing

ASM Disk Scrubbing performs a similar task to Exadata Disk Scrubbing. It searches the ASM blocks and repairs logical corruption using the mirror disks. The big difference is that ASM Disk scrubbing is run manually at disk group or file level and can be seen in V$ASM_OPERATION view and the alert_+ASM.log

### Logs and views

The alert_+ASM.log on the database node reports the command and duration.

```
Mon Feb 06 09:03:58 2017
SQL> alter diskgroup DBFS_DG scrub power low
Mon Feb 06 09:03:58 2017
NOTE: Start scrubbing diskgroup DBFS_DG
Mon Feb 06 09:03:58 2017
SUCCESS: alter diskgroup DBFS_DG scrub power low
```

## Summary

These storage maintenance tasks are not exclusive to Exadata, but rather are common to all storage vendors. A great DBA will be aware of the storage maintenance, and schedule around other high maintenance I/O activity such as RMAN backups or batch activities to keep the database running smoothly at peak performance.

### About the Author

**Andrew Meade** has been an Oracle DBA for 15 years, working in both the financial and higher education sectors. He has presented at Oracle OpenWorld and COLLABORATE. Meade is an Oracle Certified Professional for 9*i*, 10*g* and 11*g*.

# Storage Migration

**By Michelle Malcher**

The database needs storage and memory; an obvious statement. This also seems to be what the database administrator is always asking for, more disk and more memory. It is possible with Automatic Storage Management (ASM) to take advantage of any new storage technology and migrate the database by managing the ASM Disk Groups.

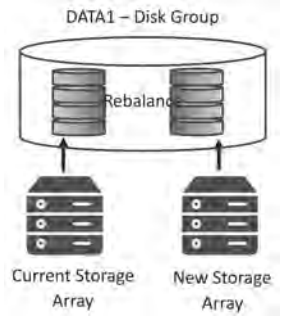
ASM Disk Groups should have disks that are the same size and it is not unheard of that size standards might change and makes it difficult to add new disk to the existing disk group. Why does the disk size need to be the same size? Simple answer: for rebalancing. If disks are different sizes, the rebalancing processes are not able to use the full disks or sometimes the rebalance is not as effective.

A few reasons to perform storage migration:

- New storage device.
- Need to move to faster disk (possible SSD).
- Maintain disk groups with same disk sizing.

Let's say we have a disk group DATA1 and it is on a storage array that will be decommissioned and they have supplied a new storage array. The zoning and mapping of the disk devices will have to be allocated, visible and permissioned for the ASM instance. If the current disk group is using disk that are 50GB in size, it is possible that the new devices are 100GB, as an example, and the disk can be added with the plan that all of the disks will not be 100GB.

The new disk is added to the disk group, and disk is rebalanced so that there is data on all of the disks.




Adding disks would be using just the ALTER DISKGROUP and a REBALANCE POWER can be set:

```
ALTER DISKGROUP data1 ADD DISK '/devices_new/disk2*'
REBALANCE POWER 4;
```

When adding disk, keep in mind that rebalancing does add an additional workload on the disk, which will mean that it is important to perform these tasks in a maintenance window. The database will still be available, so it is not an outage but it should be during a non-peak period.

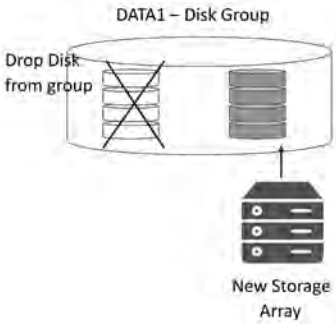
After the disk is added and rebalanced, then the disk can be dropped. Check the V$ASM_OPERATION view for information about the timing of the rebalancing.

Removing disks will leave the new storage as needed then the decommissioning process can be used to remove the storage from the ASM server.

```
ALTER DISKGROUP data1 DROP DISK '/devices_old/disk1a'
REBALANCE POWER 8;
```

View V$ASM_DISK to verify the disks that are part of the disk group.

Using ASM ADD and DROP disk from the disk group will allow for online migration to different storage. The caution is to perform the ADD, DROP and REBALANCE in non-peak times for better performance of the migration and continue to allow the database performance for users to remain constant. Using the REBALANCE POWER can also be adjusted to better performance for the migration.




## About the Author

**Michelle Malcher** is a well-known volunteer leader in security and database communities and Executive Editor of *SELECT*. Her deep technical expertise, from database to cloud appliances, as well as her senior-level contributions as a speaker, author, Oracle ACE director and customer advisory board participant have aided many corporations across architecture and risk assessment, purchasing and installation, and ongoing systems oversight.

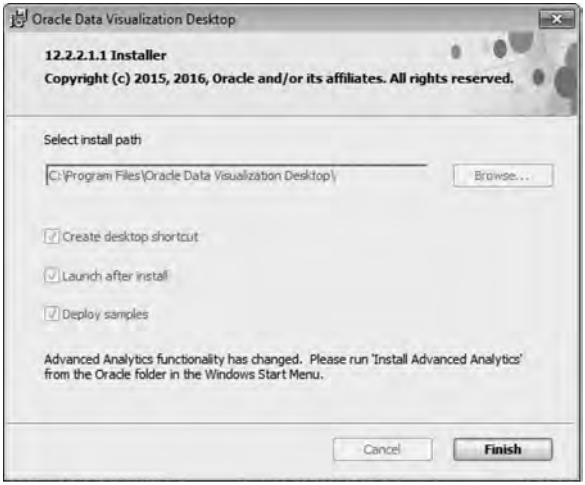
# Analyze Storage and Performance using Data Visualization Desktop (DVD) – Like a Boss

By Fahd Saeed

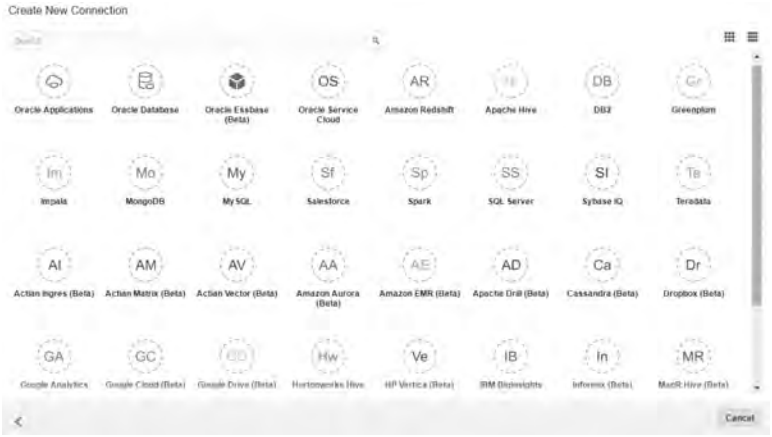
Database administrators and storage architects have a lot on their plate. With a bevy of SQL queries in their back pocket, being responsible for a multitude of databases, why wouldn't they want to create custom data visuals that tie all this information together, to either report on issues, performance, or perhaps brag about the uptimes, space and capabilities?

Oracle's newest data visualization tool, Oracle Data Visualization Desktop (DVD), allows you to quickly connect to a number of datasets within minutes and produce sweet, succulent graphical charts (like a boss). In this tip, we will take a look at quickly installing DVD, connecting DVD to SQL queries and CSV files to create visualizations that you'll be proud to show off within your organization.

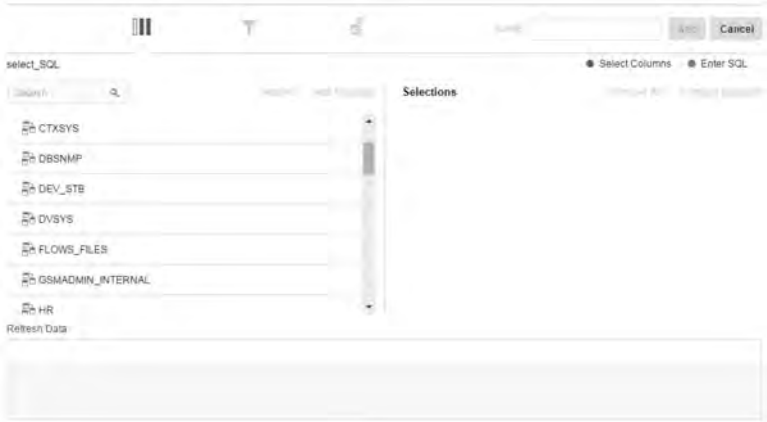

Installation of DVD is a breeze — as long as that breeze can take place on a Microsoft Windows machine running version 7, 8.1 or 10. Walk through the prompts and you should have DVD installed under "Program Files" within minutes.



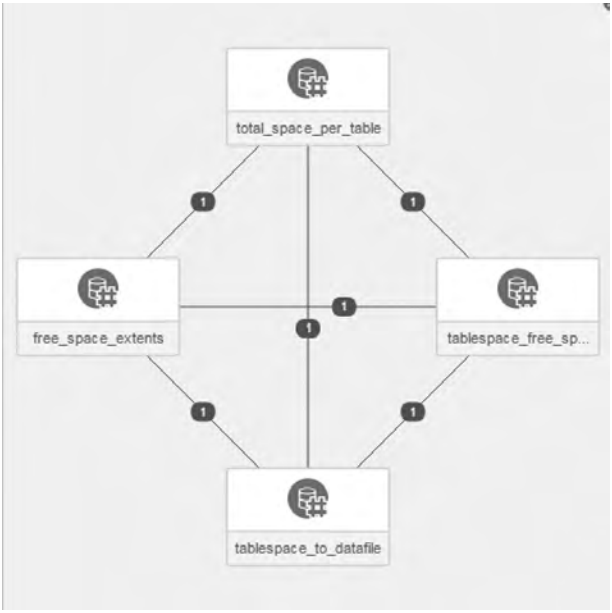

Now for the fun part: where we connect DVD to data. There is a simple File option, or you can pick one of many databases. First, let's play with the database option and choose Oracle database for our first example.



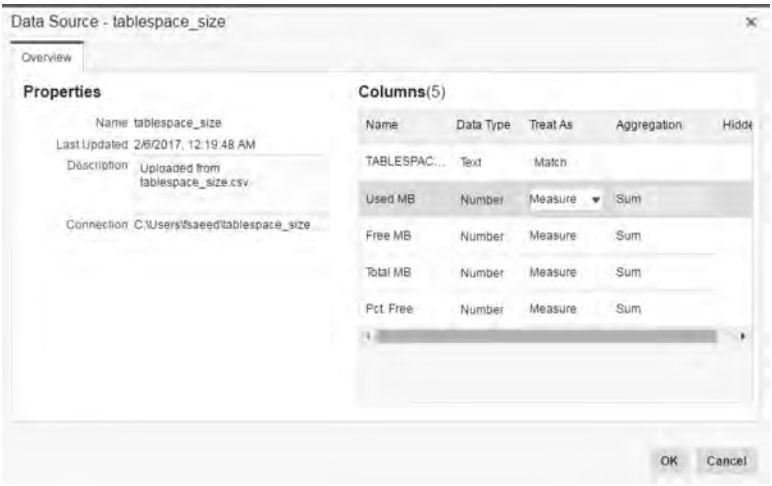
The next screen will request credentials. Here, we select the schema and tables we need to report on, or, in our case, we can simply select “SQL Query” and flop our favorite SQL into this screen to grab our unique data set. We can even click the “Result Set” to check our work before committing to this data source.



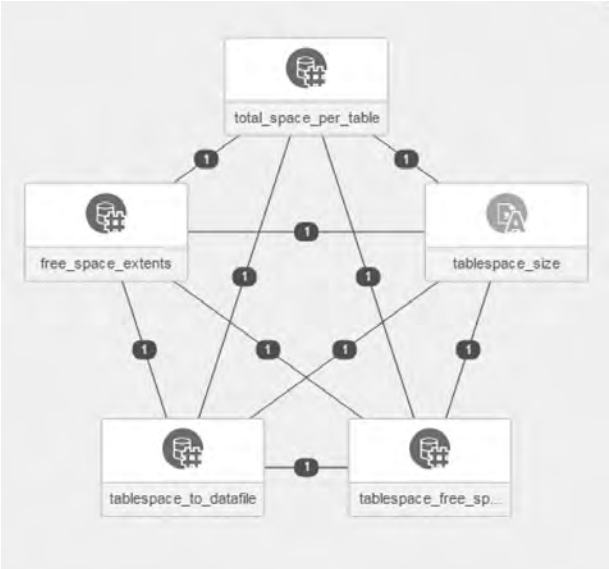

We can create one project with one or multiple data sources. Multiple data sources become effective if there are commonalities between them. This allows DVD to join the data sets together automatically. In this tip, we will add four data sources all with a common column in between them called “TableSpace_Name.” If we chose source diagram, we can see DVD was able to join them together




Four SQL Query-based data sources are good, but let’s make this slightly challenging with the addition of importing a CSV file. Imagine running a SQL in another environment, exporting a CSV result set and importing this CSV file into DVD. Once added, the CSV file might display some columns as strings rather than measures (i.e., a measure being something that is numeric and can be summed up or averaged). This can be changed by choosing “Inspect the Data Source” from the Data Sources menu. We can alter the attributes here and choose the “Treat As” value of “Measure.”




With the fifth data source added and with adjoining columns existing between them, we can check the “Source Model” and see how DVD automatically joins these data sources together.




Checking the source diagram, we can see the added CSV file fits in nicely with the existing data sources.

Now to crank up the fun. We bring it all together in a project utilizing these five data sources, selecting our columns and choosing the graph that best visualizes the result. When we drag columns from the left pane onto the canvas, DVD chooses the best graphical representation. This can be overwritten by manually choosing from the list of graphs available. The graphs can be further customized and multiple graphs can be overlayed on top of each other for aesthetics.

Notice the left pane that showcases the data sources. Each data source shows which columns that are strings and which are numeric or measures (can be aggregated, i.e., summed, averaged, etc.). The reason these data sources can all work together on this canvas is based on the joining criteria which in our case is the tablespace name but can be a session ID, schema name, etc. The canvas tab allows us to create multiple interpretations and visualizations. The results of each canvas can be exported into a nice PDF, PowerPoint and/or printed out to be shared with the masses.


Put it all together and what we have is a powerful tool that gives a colorful picture of the state of the enterprise based on your data sources. Database administrators and storage architects need not rely on gritty outputs to portray the health or wellbeing of the enterprise. They can utilize the newest data visualization tool set on the block (the DVD) to showcase our own data intelligence chops (like a boss).


## About the Author

**Fahd Saeed** is a consultant at Red Pill Analytics. Follow Fahd on Twitter (@thatFahd).

# Granting Oracle Schema Permissions (Objects Not Created Yet)

**By Mike Gangler**

Recently, we had a customer who wanted to have read-only access to all tables in a schema, even tables not created yet. Apparently in Microsoft SQL Server, there is a way to grant a user schema permissions and includes objects not created yet.

Since I am unaware of a way to do this automatically in Oracle, I decided to create a "after ddl" trigger on the schema to grant the permissions when objects are created. The challenge is that PL/SQL doesn't allow running DDL permissions; I had to create a work around for the PL/SQL and DDL challenges. The resolution was to use dbms. job_submit (Thanks, "Ask Tom") and submit a job to run the permissions.

Here is the code I used and please modify to suit your needs. For this example, I'm using IOUG as the application name/user:

1. Create a read only role first:
   Create role IOUG_READONLY;

2. Grant read only role to the user (IOUG) requiring the permissions
   grant IOUG_READONLY to IOUG;

3. Connect to application schema user (Owns objects) and create "after ddl" trigger (Sorry for the doublespace!)

```
CREATE or REPLACE TRIGGER AFTER_DDL AFTER DDL on IOUG_OBJECTS.SCHEMA
v_sysevent varchar2(25);
v_message varchar(255);
l_job number;
begin
select ora_sysevent into v_sysevent from dual;
if ( v_sysevent in ('CREATE') )
then
v_message := 'execute immediate "grant select on IOUG_OBJECTS.'||ora_dict_obj_name||' to
IOUG_READONLY";';
dbms_job.submit (l_job,replace(v_message,'"','''') ) ;
```

```
end if;
end;
/
```

Now whenever a new object gets created the role is granted via the pl/sql and dbms_job. The following is a test output:

```
Connect IOUG_OJBECTS/pw
IOUG_OBJECTS@IOUGDEV > create table foo1 (col1 varchar2(255));
Table created.
IOUG_OBJECTS@IOUGDEV > connect IOUG/pw
Connected.
IOUG@IOUGDEV > select * from IOUG_OBJECTS.foo1;
no rows selected
IOUG@IOUGDEV > desc IOUG_OBJECTS.foo1;
Name Null? Type
----------------------------------------------------------------------------------------------------------------------------
COL1 VARCHAR2(255)
select * from IOUG_OJBECTS. OBJECT_NAME
```

**NOTE:** You must put the schema name before the table name (IOUG_OBJECTS), because the grants didn't include a public synonym.

After presenting at COLLABORATE 16 about granting schema permissions, a colleague — Frank Pound — sent me the following that can also be used to perform the trigger only on "Tables:"

```
Create or replace trigger trigger_grant_dml after create on schema declare v_job number;
v_todo varchar2(200);
begin
if ora_sysevent = 'CREATE' and ora_dict_obj_type = 'TABLE'
then
v_todo:='execute immediate ''grant select, insert, update, delete on '||ora_dict_obj_name||' to
my_new_role'';';
dbms_job.submit(job=>v_job, what=>v_todo);
end if;
exception when others then null;
end;
/
```

Let me know if this works for you and big thanks to "Ask Tom" who helped me resolve the PL/SQL and DDL issue. Also, please let me know if there is an automatic way to do this in Oracle. Contact me with any questions at **mjgangler@gmail.com**.

## About the Author

**Mike Gangler** is an Oracle ACE that possesses robust database credentials with many Fortune 100 companies in positions ranging from Database Specialist / Infrastructure Architect to Senior Database Administrator.

Gangler currently serves on the board of directors of the Southeast Michigan Oracle Professionals (SEMOP) and was a charter member of the IOUG board of directors. His specialties include virtualization (VMWare, ZFS), performance tuning, both at the DB level and statement level, and Cloud Control (OEM).

# Databases and Storage – The New Frontier

By Leighton L. Nelson

As database and storage administrators and engineers, it is important to understand the effect that storage has on databases and the impact of storage performance.

It is important to understand that databases are an extension of your storage and requires a holistic view when designing and managing solutions that use the database. This tip gives an overview of the Oracle database storage landscape — past, present and future. It's not meant as in-depth technical treatment of the topic, but opens the door for more discussion and engagement about database storage.

## Evolution

Both storage and databases continue to evolve, albeit at different paces. About a decade ago, the enterprise storage of choice was the Storage Area Network, or SAN array. At about the same time, the current release of Oracle was 10*g*R1, which introduced Automatic Storage Management (ASM), provided mirroring and striping capabilities for data files.

Database sizes were typically in the hundreds of gigabytes. With the current database release of Oracle 12*c*, new storage features such as Automatic Data Optimization (ADO) and Heat Maps have been introduced to enable automatic and intelligent storage tiering within the database.

Data growth has also exploded and sizes ranging from tens to hundreds of terabytes are now common. Features such as advanced compression (with hybrid columnar compression with Oracle ZFS appliance or Exadata) can significantly reduce the amount of storage required. Additionally, when it comes to data resilience, the landscape has also significantly changed from the use of magnetic tape devices such as LTO to highly available disk-based solutions with de-duplication and replication.

We've also witnessed the proliferation of server virtualization for improving hardware efficiency. Given the evolution rate of these technologies, it becomes increasingly important for storage management professionals to keep up and arm themselves with the knowledge needed to work effectively.

## Performance

When it comes to storage performance, we're often bombarded by three specs: latency, bandwidth and IOPS. But does this tell the whole story? The short answer is "No." Each application has its own I/O requirements that are not captured by vendor benchmarks. Data access workloads (random versus sequential) and ratios, I/O sizes all play a part in storage performance. Your data warehouse on fiber-attached SAN arrays with rotating disks may give excellent performance (throughput and bandwidth), but your OLTP workload for an online application on the same array may totally suck! Can you say "latency?"

How can one truly measure storage performance? For seasoned Oracle database professionals, the answer may seem obvious: response times. But which metrics are relevant for measurement when storage is involved? The Oracle database has a well-instrumented wait interface. However, it is not perfect due to the nature of sampling.

That said, tools such as Automatic Workload Repository (AWR) and Statspack can provide a wealth of information about database performance and I/O metrics (including histograms). From a back-end storage perspective, there are tools from storage vendors that provide relevant metrics. Some vendors even provide plug-ins for Oracle Enterprise Manager Cloud Control 12*c* to enable central collection and reporting of storage metrics.

Speaking of measurement: You can find great tools to benchmark your Oracle database storage such as SLOB and fio.

## Flash

Until a few years ago, rotating disks had all but dominated enterprise storage. While not a brand-new technology, flash storage has started to proliferate the IT infrastructure in the form of SSD and, more recently, all-flash arrays. Flash provides low-latency I/O. Traditional hard-disk have service times in the milliseconds while that of flash is typically in the nanoseconds. Does that mean that you should dump all your disks for a flash-based solution? Probably not.

While flash is great at providing random I/O, spinning disk is still OK for sequential I/O. While moving all your redo logs to flash may improve response times, it may not give you the best bang for your buck, especially if there are other I/O bottlenecks. There are other issues such as durability and resilience that affect flash, but these are becoming non-issues as the technology improves.

## Software is the New Hardware

What's next for database storage? Flash technology is now becoming mainstream, and the industry is looking toward [hyper] converged infrastructure. Yes, another marketing buzzword. In a nutshell, converged infrastructure involves putting storage, network and computing resources in a single hardware unit. It's no surprise that storage has followed suit in the form of software-defined-storage (SDS) and storage virtual machines (SVM).

As an example, some vendors virtualize storage controllers in software. This allows for an agnostic storage back-end (i.e., DAS, SAN/FC, NFS). Interestingly, Oracle has continued to implement more storage related features at the database layer (e.g., Flex ASM, ACFS aka Cloud Filesystem — a POSIX compliant general purpose filesystem that includes Snapshots, Replication along with DBFS and Direct NFS).

By virtualizing hardware, storage resources businesses can become more agile (yes, another buzzword) and efficient.

## What's Next?

With the new storage frontier upon us, how does one keep up and learn more about their database storage?

The IOUG Storage SIG is a great way to keep on what's going on in the world of Oracle and database storage. There are white papers and webinars from Oracle product managers, partners and end-users. Also, check out the references section for links to white papers, articles and blogs from notable experts in the Oracle community.

## References


Kevin Closson (SLOB author): kevinclosson.wordpress.com/slob

Flashdba: "Storage for DBAs." flashdba.com/storage-for-dbas

James Morle: "Scaling Oracle 8*i*." scaleabilities.co.uk/scalingoracle8i

Frits Hoogland: fritshoogland.wordpress.com

Michael Webster: longwhiteclouds.com/oracle

Kyle Hailey: "Lies, Damned Lies, and I/O Statistics." oraclerealworld.com/lies-damned-lies-and-io-statistics

Lonny Niederstadt: "sql.sasquatch." sql-sasquatch.blogspot.com

## About the Author

**Leighton Nelson** works as an instructor at Delphix, providing education to customers for Delphix Virtualization and Masking products. He has more than a decade of experience working in various data roles, including database administrator, developer and consultant with Oracle, MS SQL Server and MySQL among others.

Nelson holds a Bachelor of Science degree in Computer Science, Oracle 11*g* RAC Expert, Oracle Database 12*c* and VMware VCA-DCV certifications as well as Oracle ACE recognition. He is also a co-author of *Expert Oracle Enterprise Manager 12*c, published by Apress.

# Optimize Oracle Database Storage with Storage Tiering

By Kai Yu

## Storage Tiering: Performance vs. Cost

While SSD-based storage can significantly improve the I/O performance and reduce the database performance bottlenecks on storage I/Os, the cost of having the entire database stored in the SSD storage can be relatively high, especially when the data volume is in exponential growth. However, for many databases, it is quite common that only a small portion of data is frequently accessed (hot data) and the rest of data is relatively idle (cold data), for example, those historical data or archive data. The I/O performance on those less frequently accessed data has less impact on the database performance.

To understand how the I/O performance on hot data impacts the database performance, we have performed studies on database performance vs. the distribution of data between SSD storage and HDD storage. The studies were conducted on seven different data distributions ranging from storing all database objects in HDD storage to storing all database objects in SSD storage as follows:

| Case# | HDD Storage | SSD storage |
|---|---|---|
| Config#1 | All database objects | None |
| Config#2 | all other objects not in SSD | All redo logs, undo tablespace, and temporary tablespace |
| Config#3 | all other objects not in SSD | all user tables indexes, all redo logs, undo tablespace, and temporary tablespace |
| Config#4 | all other objects not in SSD | One most active table, all user table indexes, all redo logs, undo tablespace, and temporary tablespace |
| Config#5 | all other objects not in SSD | Two most active tables, all user table indexes, all redo logs, undo tablespace, and temporary tablespace |

| Case# | HDD Storage | SSD storage |
|---|---|---|
| Config#6 | All remaining database objects not in SSD | Four most active tables, all user table indexes, all redo logs, undo tablespace, and temporary tablespace |
| Config#7 | None | All user schema objects, all redo logs, undo tablespace, and temporary tablespace |

*Table 1: Test configurations of data distributions between SSD and HDD storage*

The performance baseline was based on the Config1 with all database objects stored in HDD storage. All other six configurations were used to compare with this base line. The performance results were based on a 500 GB database with some TPCC like transactional application workloads. The SSD storage consisted of 8 PCIe based SSDs in RAID 10 configuration and the HDD storage consisted of 48x10k rpms Hard Disks in RAID 10 configuration. Figure 1 shows the database transaction throughputs of the seven configurations:

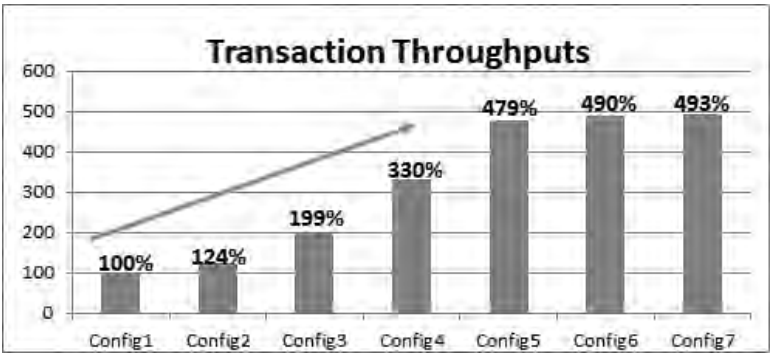



*Figure 1: Transaction Throughputs vs. Data Distributions between two tiers of storage*

The database average transaction response time comparisons on these seven configurations:

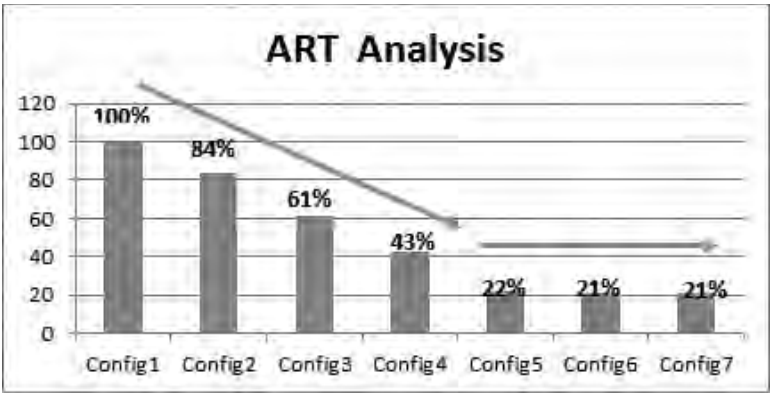



*Figure 2: Transaction Average Response Time (ART) vs. Data Distribution between two tiers of storage*

Both Figure 1 and Figure 2 show that by storing the hot data (indexes plus two hot tables as in Config5) in the SSD storage we can achieve the similar database performance as we achieve by storing all the data in SSD storage (refer to Config7). In Config5, this hot data counts less than one-third of total database capacity. This example shows that the tiered storage allows us to achieve a high performance by storing a small portion of data in the fast SSD storage (Tier 1) and storing the rest of the data in slow HDD storage (Tier 2).

## Implementing Storage Tiering for Oracle Database

The key to storage tiering is how to select the performance sensitive frequently accessed hot data and put them into the tier 1 SSD storage and leave the rest of less performance sensitive and less frequently accessed the data to the tier 2 or tier 3 HDD (Hard Disk Drive) storage. There are a few options to implement the storage tiering:

- **Built-in storage tiering in the storage product.** Many storage vendors provide the storage tiering feature with their storage product. This storage tiering feature tracks the usage of the data blocks in the storage and moves the data blocks and pages of data between the different tiers within the storage array based on predefined rules and how data has been actively accessed.

- **Manual method.** Manually move the database objects to a proper storage tier by DBAs. DBAs can move the data objects to the proper tier of the storage based on its performance requirement, data access pattern and capacity of the tiered storage.
- **Information management feature of application or database product.** If the tiered storage consists of multiple storage arrays, the storage tiering needs to be independent from a particular physical storage array so that it can manage the data move among the multiple physical storage arrays. In this case, it is ideal to have the storage tiering method implemented on the database level, instead on the physical storage level.

For this reason, we look at the new database features introduced in Oracle 12*c* called Oracle Information Lifecycle Management (ILM) and Automatic Data Optimization (ADO). These two fit well for implementation of storage tiering on the database level. The rest of article will discuss how to leverage them for storage tiering.

## How to Leverage Oracle ILM for Storage Tiering

Oracle ILM consists of two parts:

1. Heat Map, which tracks and stores system-generated data usage statistics about the database segments such as tables, partitions and indexes.
2. Automatic Data Optimization (ADO) that allows DBAs to create policies, and automate actions based on those policies.

In order to use Oracle ILM/ADO to implement storage tiering, DBAs need to define the proper polices for ADO to automate the actions that move data between the tiers of storages, like this:


Setup an ADO policy for data movement:

For example, to define the policy that moves the ORDER_HISTORY table from the current tablespace T1DATA in Tier 1 storage to T2DATA tablespace in Tier 2 storage, we can create a policy like this:

```
SQL> ALTER TABLE ORDER_HISTORY ILM ADD POLICY TIER TO T2DATA;
```

The policy also can be defined for on a partition level, for example to move a partition ORD2 to T3DATA tablespace in the Tier 3 storage:

```
SQL> ALTER TABLE ORDER_HISTORY MODIFY PARTITION ORD2 ILM ADD POLICY TIER TO T3DATA;
```

When these policies are defined, Oracle ADO will trigger the action once the tablespace utilization meets the certain thresholds which are tracked by two ILM ADO parameters: TBS PERCENT USED and TBS PERCENT FREE column in dictionary view DBA_ILMPARAMETER:

```
SQL> SELECT NAME, VALUE FROM DBA_ILMPARAMETER WHERE NAME LIKE 'TBS%'
NAME                      VALUE
------------------------  -----
TBS PERCENT USE                    85
TBS PERCENT FREE                   25
```

TBS PERCENT USED indicates the percentage threshold of the tablespace full when the table starts moving to the new tablespace. TBS PERCENT FREE indicates the percentage threshold of tablespace full when the table stops moving to the new tablespace. In the example above, ORDER_HISTORY will be moved to T2DATA tablespace once its current tablespace T1DATA has reached 85 percent full, and it will stop moving when the T2DATA tablespace space usage is reduced to 25 percent full.

These threshold numbers can be customized by the *CUSTOMIZE_ILMP procedure.* For example, we can alter the threshold numbers to 90% and 15% respectively by calling the *CUSTOMIZE_ILM* procedure like this:

```
SQL> BEGIN
  2 DBMS_ILM_ADMIN.CUSTOMIZE_ILM(DBMS_ILM_ADMIN.TBS_PERCENT_USED, 90);
  3 DBMS_ILM_ADMIN.CUSTOMIZE_ILM(DBMS_ILM_ADMIN.TBS_PERCENT_FREE, 15);
  4 END;
  5 /
```

The usage statistics such as the percentage used and percentage free of all tablespace are tracked in the database system. We can run the following SQL query to get the statistics in an SQLPLUS session.

```
SQL> SELECT A.TABLESPACE_NAME TBS_NAME,
 ROUND ((NVL (B.BYTES_FREE, 0) / A.BYTES_ALLOC) * 100, 2) PCT_FREE,
 100 - ROUND ((NVL (B.BYTES_FREE, 0) / A.BYTES_ALLOC) * 100, 2) PCT_USED
 FROM (SELECT F.TABLESPACE_NAME, SUM (F.BYTES) BYTES_ALLOC,
 SUM (DECODE (F.AUTOEXTENSIBLE,'YES', F.MAXBYTES,'NO', F.BYTES)) MAXBYTES
 FROM DBA_DATA_FILES F
 GROUP BY TABLESPACE_NAME) A,
 (SELECT F.TABLESPACE_NAME, SUM (F.BYTES) BYTES_FREE
 FROM DBA_FREE_SPACE F
 GROUP BY TABLESPACE_NAME) B
 WHERE A.TABLESPACE_NAME = B.TABLESPACE_NAME (+);
```

With Oracle ILM and ADO, the condition for the table move between tablespaces in different tiers is based on the tablespace capacity utilization thresholds defined by TBS PERCENT USE and TBS PERCENT FREE parameters.


## About the Author

**Kai Yu** is a senior principal architect in Dell EMC Enterprise Solutions Engineering. Yu spent more than two decades architecting and implementing various Oracle solutions by specializing in Oracle RAC database, virtualization/ Cloud, IT End to End Infrastructure including servers, Networking and Storage. He has given more than 130 presentations in various IT conferences such as Oracle OpenWorld, IOUG/OAUG Collaborates and IEEE HPCC conference. He has been an Oracle ACE Director since 2010 and has shared all his technical articles and conference presentations on his Oracle blog: **kyuoracleblog.wordpress.com**.

# Temporary I/O Poisoning of Storage

**By Mike Ault**

Many times you will see that the temporary tablespace is the source of a majority of your I/O. If you do massive temporary actions, such as sorts in the 1-2 gigabyte range, this may be perfectly normal. However, if your temporary actions are less than 512 megabytes in size, you may be experiencing temporary I/O poisoning of your storage.

In Oracle version 10 or so, we were given the PGA_AGGREGATE_TARGET setting and automated temporary segment management. The parameter _PGA_MAX_SIZE was provided to set the most amount of memory to be allocated per session. Depending on the release, this parameter was set to either 256 or 512 megabytes.

If shared server (think connection pooling) is turned on by either setting the DISPATCHERS or SHARED_SERVERS parameters, then PGA_AGGREGATE will not be used for parallel processes (read parallel query or parallel DML). Instead, the old parameters (SORT_AREA_SIZE, HASH_AREA_SIZE and other related parameters) will be used. Since SORT_AREA_SIZE defaults to 64K, and most DBAs these days aren't even aware the old parameters exist, let alone how to set them, this results in what is known as out-of-band (OOB) temp segments.

Any temporary segment less than 512 megabytes should be handled by the PGA_AGGREGATE_TARGET and automatic temporary segment control, if the PGA_AGGREGATE_TARGET setting is proper. There are two sections in the AWR report that will tell us if we have OOB temp segments and if PGA_AGGREGATE_TARGET is set properly. The PGA Histogram shows for ranges of segment sizes how many one-pass and multi-pass segments were written to disk.

The PGA Target histogram tells us for different settings of PGA_AGGREGATE_TARGET what the projected effects would be on PGA processing. If the over-allocation column has values that reach beyond the current setting (size factor 1), then the PGA_AGGREGATE_TARGET is undersized. However, if the over-allocation column has no values or the greater than zero values stop prior to the size factor setting of 1, then the PGA_AGGREGATE_TARGET setting is most likely correct and something else is forcing OOB temp segments.

Oracle by default sets up one dispatcher, usually with the postfix of XDB, and the shared server setting defaults to 1 in some versions. This was done by Oracle because they assumed you would use their tools and their tools will connect to the XDB dispatcher. In the years of analysis done with thousands of AWR and statspacks, I have never seen this utilized.

Set the DISPATCHERS to NULL and SHARED_SERVERS to zero (0). The SHARED_SERVER parameter will not be displayed since 1 is the default value in earlier releases. In Oracle11 and later releases it defaults to 0. By turning off shared server, many of the OOB temporary actions can be mitigated.

If specific users use large temporary areas for index building, table creation, large hash joins or other temporary segment intensive operations, consider using a login trigger to reset their sort and hash area sizes accordingly as setting them manually via a trigger or command over-rides PGA_AGGREGATE_TARGET settings.


### About the Author

**Mike Ault** has worked with a variety of industries utilizing Oracle, both in-house and as a consulting talent, since 1990. He is currently a Business Solutions Architect specializing in Oracle and FlashSystems for the Storage and SDI Solutions group at IBM.

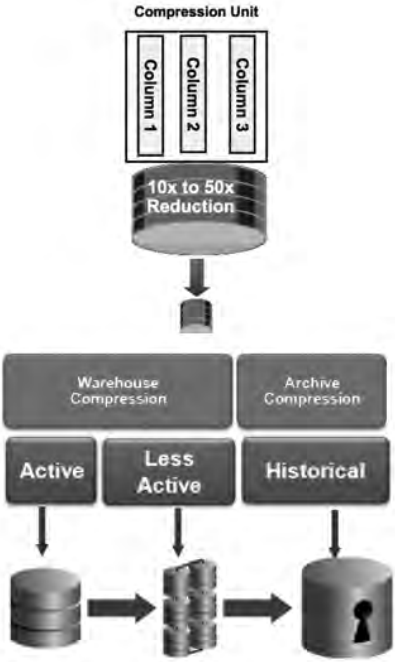
Ault has published more than two dozen Oracle-related books, and he has written articles for Oracle, IOUG *SELECT*, DBMS, Oracle Internals and several other database-related magazines.

# Hybrid Columnar Compression with the Oracle ZFS Storage Appliance

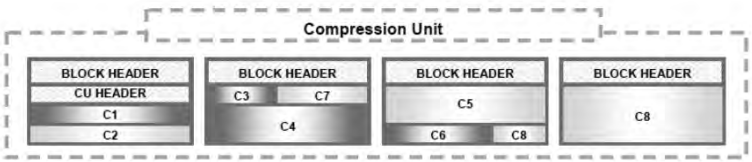
By Rawley Burbridge

The Oracle ZFS Storage Appliance is an enterprise multiprotocol storage platform that offers extensive functionality for Oracle Database. With an advanced operating system employing Hybrid Storage Pools, and cost-effective yet powerful controllers, the Oracle ZFS Storage Appliance delivers high performance at a lower price point than major competitors. Since Oracle ZFS Storage Appliance is co-engineered with Oracle software, including Oracle Database, it includes features such as Oracle Intelligent Storage Protocol and Hybrid Columnar Compression which provide optimizations that are only available when using Oracle on Oracle.

Oracle Hybrid Columnar Compression (HCC) is an Oracle Database compression technology which produces the highest level of data compression, while also improving performance through reduced I/O. Hybrid Columnar Compression is ideally suited for OLTP or Data Warehouse applications where the data has no or few modifications. Customers have reported compression ratios of 6x to 50x by using HCC with their Oracle Databases.




These high compression ratios are achieved by the unique approach of using a combination of both row and columnar methods for storing data, which optimizes compression while minimizing the performance penalties of a pure columnar format. Data is stored in a logical construct called the compression unit, which stores 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.



There are four different HCC algorithms that can be applied to a database table or partition. Two are suitable Data Warehouse access profiles, while the other two are suitable for Archive access profiles with inactive datasets:

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

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

```
CREATE TABLE MY_TABLE TABLESPACE TEST COMPRESS FOR QUERY HIGH AS SELECT * FROM EXP;
```

This statement creates a table named "MY_TABLE" in a tablespace named "TEST", sets the table compression attribute to "QUERY HIGH", and loads the table with the rows from table "EXP". The result is a table with data compressed using the HCC Query High algorithm.

It is important to understand the potential benefits of HCC prior to implementing it, therefore it is recommended to run the Oracle Advanced Compression Advisor tool (DBMS_COMPRESSION), which is a PL/SQL package included with Oracle Database. This tool can be used to estimate potential storage savings by using HCC, based on analysis of a sample of data. The actual HCC algorithms are used to compress a subset of target data to judge the effectiveness of each option. The Advanced Compression Advisor provides a good estimate of the actual results that may be obtained after implementing HCC on a dataset.

There is no extra licensing necessary to use Hybrid Columnar Compression, however it does require Oracle Database Enterprise Edition. Hybrid Columnar Compression also requires Oracle storage be used, including Oracle ZFS Storage Appliance. Oracle co-engineering produces many benefits that ensure that Oracle software runs best on Oracle hardware, Hybrid Columnar Compression is just one example.

## About the Author

**Rawley Burbridge** is a seasoned technologist with expertise in enterprise infrastructure and cloud implementations, currently acting as a pre-sales solutions architect at Oracle. He spends his time helping customers design and implement complex enterprise solutions utilizing Oracle technologies. Burbridge makes his home in Nixa, Missouri.

# Optimizing Database Performance Using Oracle ZFS Storage Analytics

By Rawley Burbridge

The Oracle ZFS Storage Appliance is architected to accelerate Oracle Database and applications while reducing deployment and management risks. Oracle's co-engineering of software and hardware has enabled innovative features to be created which ensure Oracle applications run best on Oracle hardware. For example, the Oracle ZFS Storage Appliance is the only network attached storage (NAS) system with container-level visibility for Oracle Database 12*c* and a direct communication path with Oracle Databases so it can automatically adapt to the changing needs of the database.

The Oracle Intelligent Storage Protocol (OISP) is a mechanism by which Oracle Database 12*c* can pass cues to the Oracle ZFS Storage Appliance over the Direct NFS client. Without OISP, the best practice when using a NAS system for database recommends creating multiple NFS shares for each of the major database file types (data files, control files, online redo log files, etc.).

This allows each share to be optimally tuned for the different record size and logbias settings for the workload profile of each file type. Performance can be enhanced by tuning, but it is a manual effort. With OISP, attributes such as the type of database file that the I/O request is targeting, the record size of the file and the identity of the database issuing the I/O request are all passed to the Oracle ZFS Storage Appliance. This allows the database shares to be consolidated and the storage adjusts share parameters automatically for each of the different file types.

The Oracle ZFS Storage Appliance includes DTrace Analytics, which is an advanced facility which can be used to record and graph a variety of statistics in real-time and record it for later viewing. Analytics can be used for long term monitoring and short term analysis, and the depth of statistics available can aid in identifying and troubleshooting performance issues. With the addition of OISP, Oracle ZFS Storage Appliance Analytics provides drill down statistics on a per-database basis, including at the pluggable database (PDB) level in Oracle Multitenant environments.

The following steps demonstrate how to setup a monitor which shows NFSv4 operations per database instance:

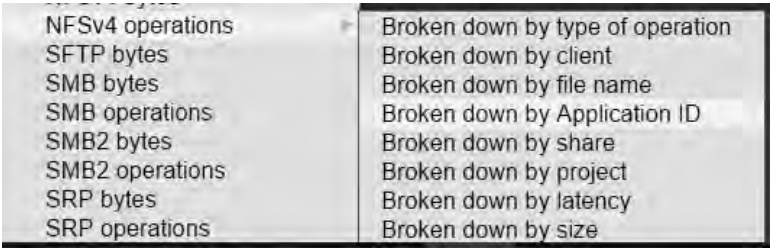
## Navigate to the Analytics section of the ZFS Storage Appliance Browser User Interface (BUI)

Click the *Analytics* link in the top right of the Oracle ZFS Storage Appliance BUI.

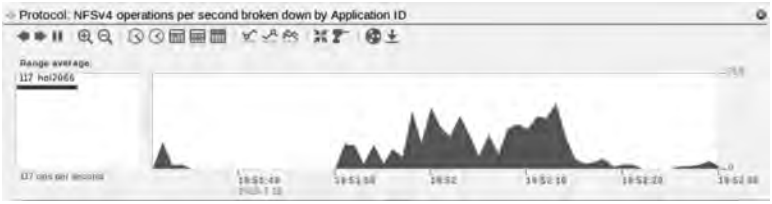

## Add a Statistic

Click the (+) icon next to *Add statistic* along the left side of the BUI.


Highlight NFSv4 operations and then select “Broken down by Application ID.”




This will create an Analytics worksheet which displays the NFSv4 operations for each of the database instances on the system. In this example, a single database called “hol2066” is running.



## About the Author

**Rawley Burbridge** is a seasoned technologist with expertise in enterprise infrastructure and cloud implementations, currently acting as a pre-sales solutions architect at Oracle. He spends his time helping customers design and implement complex enterprise solutions utilizing Oracle technologies. Burbridge makes his home in Nixa, Missouri.

# Notes

# Notes

# SELECT

*The intelligence hub of the Oracle technology and database community*

## The new *SELECT* is a dynamic intelligence hub. **Check it out at SELECT.IOUG.org >**

To meet the needs of today's readers, *SELECT* has moved from a traditional quarterly publication to a dynamic intelligence hub where timely and relevant content is published on a continual basis. That means **more** of the latest technical content for IOUG members regularly throughout the year.

Want to be part of the *SELECT* community? We are looking for writers and reviewers of *SELECT* content. Contact *SELECT* Managing Editor Mike Carrozzo at **mcarrozzo@ioug.org** for guidelines.

IOUG
independent oracle users group

IOUG
independent oracle users group

**Independent Oracle Users Group**
330 N. Wabash Ave.
Chicago, IL 60611 USA
Phone: +1.312.245.1579
Fax: +1.312.527.6785
email: ioug@ioug.org