An Oracle Technical White Paper

October 2013

Configuring Multipathing for Oracle Linux and the Oracle ZFS Storage Appliance

Executive Overview

In any highly available service configuration, there will be several components in the data path that are duplicated in order to provide resiliency and continuity of data access. These duplicated or redundant components are used to avoid any single point of failure. In this redundant setup, the server must be configured so that it is aware of these multiple components or paths so that they can be used efficiently and effectively. Consistent names and ownership are key to this successful redundancy.

Oracle Linux provides a configurable multipathing solution based on Linux DM-Multipath (also known as device-mapper-multipath.) DM-Multipath requires configuration from the base up. This paper will show how to configure DM-Multipath for use with the Oracle ZFS Storage Appliance family in an iSCSI environment so that volumes can be presented in a consistent way.

In certain circumstances, it is necessary to establish ownership of these volumes other than through the standard Linux defaults – such as when the Oracle ZFS Storage Appliance-presented LUNs will be used with Oracle Automatic Storage Manager (ASM). In cases such as this, it is possible to use the Oracle Linux device manager UDEV to automatically change ownership of the volumes to the appropriate owner at boot time, so that databases can start automatically without administrator intervention.

Introduction

The Oracle ZFS Storage Appliance family can provide highly resilient storage to a large number of different operating systems, but the infrastructure to allow resilient access to this storage depends on configuration of the Oracle ZFS Storage Appliance and the hosts to which access has been granted.

This document shows how to implement this resilient access through Oracle Linux's DM-Multipath facility, and how you can use additional facilities present through DM-Multipath and Oracle Linux's UDEV to provide consistent naming and ownership of the LUNs presented by the Oracle ZFS Storage Appliance.

This document assumes the example configuration shown in the following figure. Two Oracle Linux servers, one for Release 5 and one for Release 6, are connected to three data networks and an administration network.

An Oracle ZFS Storage Appliance is deployed, attached to each of the three data networks and to the administration network.

The three data networks are dedicated to the task of providing the multiple redundant paths required for a highly available system.

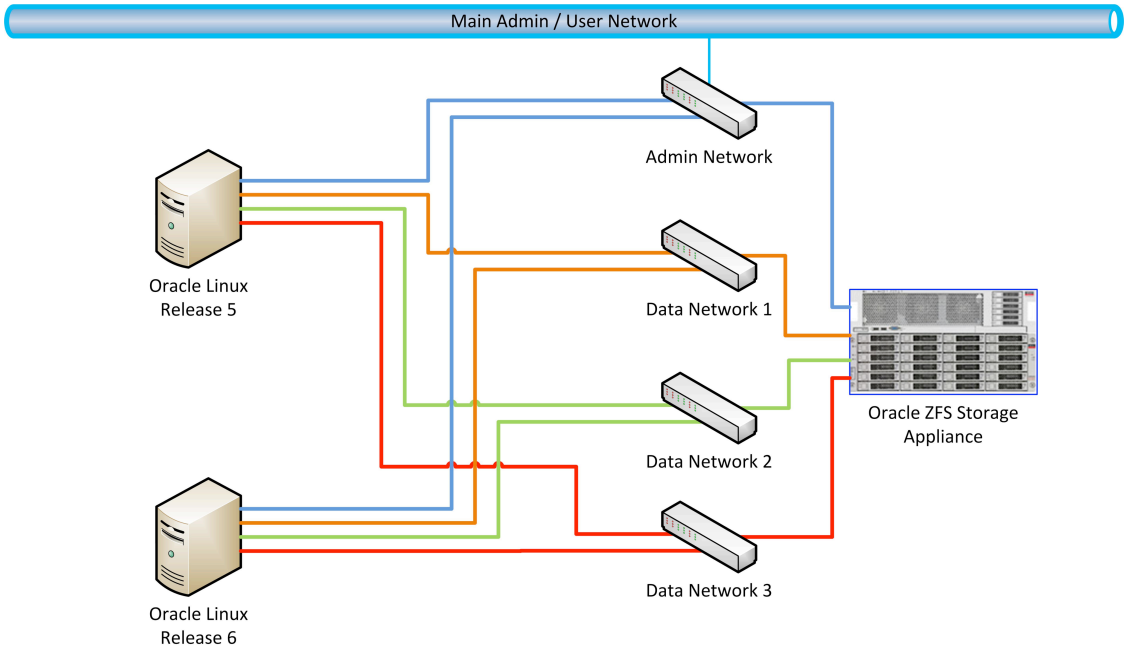

Figure 1. Sample network diagram for multipathed Oracle Linux instances and Oracle ZFS Storage Appliance

DM-Multipath and Oracle Linux

DM-Multipathing is the facility offered by Oracle Linux by which multiple paths to storage can be automatically managed and maintained through failure and failure resolution with no or minimal administrator intervention.

The block model in figure 2 shows a functional schematic for the data path stack.

DM-Multipath sits between the volume managers – such as Linux Logical Volume Manager (LVM) and Oracle Automatic Storage Management – and the storage providers (in this case, the SCSI layer). DM-Multipath provides a level of abstraction from the storage providers to hide the redundancy required for resilient configurations.

In the example shown, a LUN presented from the Oracle ZFS Storage Appliance is represented by three device nodes at the SCSI layer, and then shows up as three SCSI disk devices in `/dev` on the Linux server.

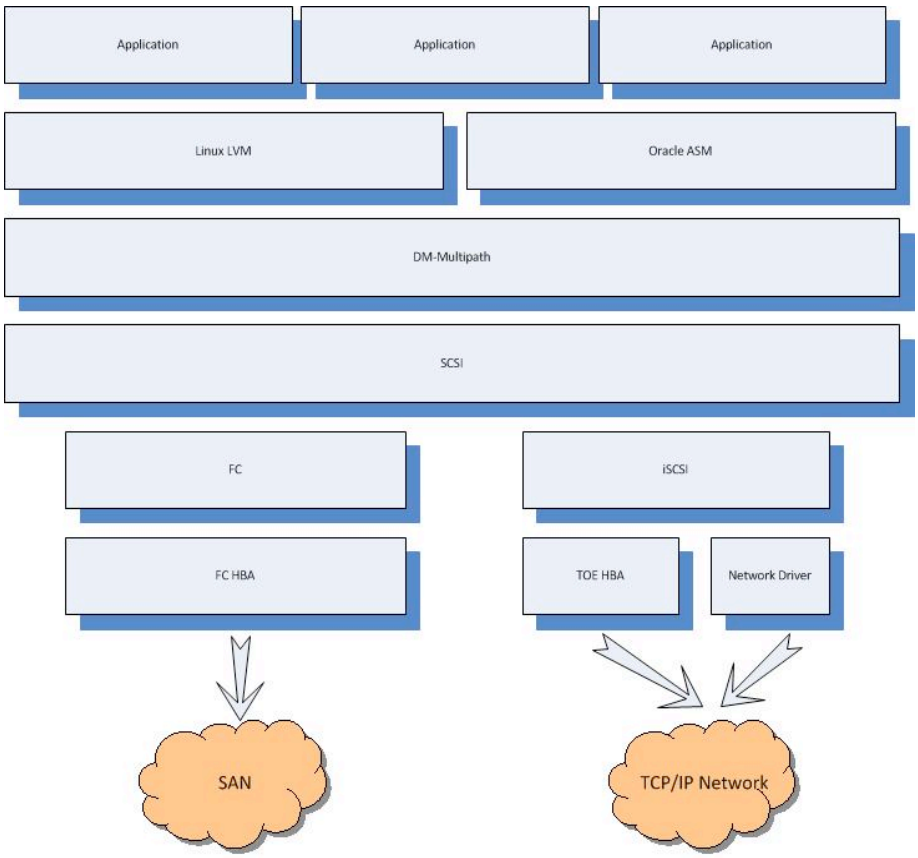

Figure 2. Functional perspective of DM Multipathing

Installing DM-Multipath

DM-Multipath may or may not be installed at initial system installation time, depending on which installation package choice is made. If needed, it is not difficult to install DM-Multipath if either the installation media is still present or the Oracle Linux server has been configured to allow access to Oracle's Public Yellowdog Updater Modified (YUM) server. Oracle's public YUM server can be used post-server-installation for package updating and amendment.

Ensure that the Oracle Linux server has the correct configuration files in place to allow the server to contact the Oracle public YUM server.

The first example shown is for Oracle Linux Release 6.

```
[root@ol-r6 ~]# cd /etc/yum.repos.d
[root@ol-r6 yum.repos.d]# ls -l
total 0
[root@ol-r6 yum.repos.d]# wget http://public-yum.oracle.com/public-yum-
ol6.repo
--2013-02-25 12:50:32--  http://public-yum.oracle.com/public-yum-ol6.repo
Resolving public-yum.oracle.com... 141.146.44.34
Connecting to public-yum.oracle.com|141.146.44.34|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2411 (2.4K) [text/plain]
Saving to: “public-yum-ol6.repo”

100%[======================================>] 2,411       --.-K/s   in 0.001s

2013-02-25 12:50:32 (3.80 MB/s) - “public-yum-ol6.repo” saved [2411/2411]

[root@ol-r6 yum.repos.d]# vi public-yum-ol6.repo
//
// Ensure that the first entry, [ol6_latest] ends with the line enabled=1
//
```

The second example shown is for Oracle Linux Release 5.

```
[root@ol-r5 /]# cd /etc/yum.repos.d
[root@ol-r5 yum.repos.d]# ls -l
total 0
[root@ol-r5 yum.repos.d]# wget http://public-yum.oracle.com/public-yum-
el5.repo
--2013-02-25 13:04:47--  http://public-yum.oracle.com/public-yum-el5.repo
Resolving public-yum.oracle.com... 141.146.44.34
Connecting to public-yum.oracle.com|141.146.44.34|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4220 (4.1K) [text/plain]
Saving to: `public-yum-el5.repo'

100%[======================================>] 4,220       --.-K/s   in 0.004s

2013-02-25 13:04:47 (1.05 MB/s) - `public-yum-el5.repo' saved [4220/4220]

[root@ol-r5 yum.repos.d]# vi public-yum-el5.repo
//
```

```
// Ensure that the first entry, [el5_latest] ends with the line enabled=1
//
```

The webpage at http://public-yum.oracle.com gives instructions for each particular Oracle Linux release.

At this point, the steps are the same for Oracle Linux Releases 5 and 6. First, search for the correct package name.

```
[root@ol-r6 yum.repos.d]# yum search multipath
Loaded plugins: refresh-packagekit, security
============================== N/S Matched: multipath
==============================
device-mapper-multipath.x86_64 : Tools to manage multipath devices using
                               : device-mapper
device-mapper-multipath-libs.x86_64 : The device-mapper-multipath modules and
                                    : shared library

  Name and summary matches only, use "search all" for everything.
```

Next, thanks to the dependency resolution built into YUM, it is only necessary to specify the Tools package – `device-mapper-multipath`. YUM will prompt for user verification that the operations it will take are approved, and manages any additional packages that may be required (in this case, the `device-mapper-multipath-libs` package).

```
[root@ol-r6 yum.repos.d]# yum install device-mapper-multipath
Loaded plugins: refresh-packagekit, security
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package device-mapper-multipath.x86_64 0:0.4.9-56.el6_3.1 will be
installed
--> Processing Dependency: device-mapper-multipath-libs = 0.4.9-56.el6_3.1
for package: device-mapper-multipath-0.4.9-56.el6_3.1.x86_64
--> Processing Dependency: libmultipath.so()(64bit) for package: device-
mapper-multipath-0.4.9-56.el6_3.1.x86_64
--> Running transaction check
---> Package device-mapper-multipath-libs.x86_64 0:0.4.9-56.el6_3.1 will be
installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                       Arch     Version             Repository    Size
```

```
================================================================================
Installing:
 device-mapper-multipath        x86_64   0.4.9-56.el6_3.1    ol6_latest    96 k
Installing for dependencies:
 device-mapper-multipath-libs   x86_64   0.4.9-56.el6_3.1    ol6_latest   158 k

Transaction Summary
================================================================================
Install       2 Package(s)

Total download size: 254 k
Installed size: 576 k
Is this ok [y/N]: y
Downloading Packages:
(1/2): device-mapper-multipath-0.4.9-56.el6_3.1.x86_64.r |  96 kB     00:00
(2/2): device-mapper-multipath-libs-0.4.9-56.el6_3.1.x86 | 158 kB     00:00
--------------------------------------------------------------------------------
Total                                           104 kB/s | 254 kB     00:02
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : device-mapper-multipath-libs-0.4.9-56.el6_3.1.x86_64         1/2
  Installing : device-mapper-multipath-0.4.9-56.el6_3.1.x86_64              2/2
  Verifying  : device-mapper-multipath-0.4.9-56.el6_3.1.x86_64              1/2
  Verifying  : device-mapper-multipath-libs-0.4.9-56.el6_3.1.x86_64         2/2

Installed:
  device-mapper-multipath.x86_64 0:0.4.9-56.el6_3.1

Dependency Installed:
  device-mapper-multipath-libs.x86_64 0:0.4.9-56.el6_3.1

Complete!
[root@ol-r6 yum.repos.d]# chkconfig multipathd on
```

Now that DM-Multipath is installed, the next step is configuration for the particular setup.

Configuring DM-Multipath in Oracle Linux

Note: The following procedure is applicable for both Oracle Linux Release 5 and Release 6 in this form.

DM-Multipath uses a main configuration file `/etc/multipath.conf` which, due to its highly subjective content, is not created upon installation. The first step in configuration is to create this file. A tool exists to do this semi-automatically. However, since the resultant configuration file, which is easily deployed, still needs editing prior to implementation, you can choose to create the file directly in an editor.

The content of the `/etc/multipath.conf` file should have, at minimum, the following:

```
devices {
       device {
              vendor               "SUN"
       }
}
```

If multipath-capable devices are already configured on the Oracle Linux server, these should be left intact and the above line added to the `devices` declaration. If there are additional Oracle ZFS Storage Appliance products defined, it may be necessary to add a `product` declaration in addition to the `vendor` declaration shown. See the manual page for `multipath.conf(5)` for additional information.

Once the specific LUN details presented by the Oracle ZFS Storage Appliance are known, the `/etc/multipath.conf` file will need further editing to customize the configuration.

With the skeleton configuration file in place, the iSCSI subsystem can now be started. The startup display will provide the specific LUN configuration details.

```
[root@ol-r5 ~]# service iscsid start
Starting iSCSI daemon:                                 [  OK  ]
root@ol-r5 ~]# iscsiadm -m discovery -t sendtargets -p 192.168.1.13
192.168.2.13:3260,2 iqn.1986-03.com.sun:02:bfe9431c-3bba-c1f2-d72f-
a0829cc2d637
192.168.3.13:3260,3 iqn.1986-03.com.sun:02:bfe9431c-3bba-c1f2-d72f-
a0829cc2d637
192.168.4.13:3260,4 iqn.1986-03.com.sun:02:bfe9431c-3bba-c1f2-d72f-
a0829cc2d637
[root@ol-r5 ~]# iscsiadm -m node --login
Logging in to [iface: default, target: iqn.1986-03.com.sun:02:bfe9431c-3bba-
c1f2-d72f-a0829cc2d637, portal: 192.168.3.13,3260] (multiple)
Logging in to [iface: default, target: iqn.1986-03.com.sun:02:bfe9431c-3bba-
c1f2-d72f-a0829cc2d637, portal: 192.168.2.13,3260] (multiple)
Logging in to [iface: default, target: iqn.1986-03.com.sun:02:bfe9431c-3bba-
c1f2-d72f-a0829cc2d637, portal: 192.168.4.13,3260] (multiple)
Login to [iface: default, target: iqn.1986-03.com.sun:02:bfe9431c-3bba-c1f2-
d72f-a0829cc2d637, portal: 192.168.3.13,3260] successful.
Login to [iface: default, target: iqn.1986-03.com.sun:02:bfe9431c-3bba-c1f2-
d72f-a0829cc2d637, portal: 192.168.2.13,3260] successful.
Login to [iface: default, target: iqn.1986-03.com.sun:02:bfe9431c-3bba-c1f2-
d72f-a0829cc2d637, portal: 192.168.4.13,3260] successful.
[root@ol-r5 ~]# service multipathd start
Starting multipathd daemon:                            [  OK  ]
```

```
[root@ol-r5 ~]# multipath -ll
3600144f0c532bda00000512a754e0001 dm-2 SUN,ZFS Storage 7420
size=4.0G features='1 queue_if_no_path' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=1 status=enabled
| `- 4:0:0:0 sdb 8:16 active ready running
|-+- policy='round-robin 0' prio=1 status=enabled
| `- 5:0:0:0 sdc 8:32 active ready running
`-+- policy='round-robin 0' prio=1 status=enabled
  `- 3:0:0:0 sdd 8:48 active ready running
3600144f0c532bda00000512a75600002 dm-3 SUN,ZFS Storage 7420
size=8.0G features='1 queue_if_no_path' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=1 status=active
| `- 5:0:0:1 sdf 8:80 active ready running
|-+- policy='round-robin 0' prio=1 status=enabled
| `- 3:0:0:1 sdg 8:96 active ready running
`-+- policy='round-robin 0' prio=1 status=enabled
  `- 4:0:0:1 sde 8:64 active ready running
```

This last command, `multipath -11`, shows the details required to provide symbolic names to the volumes. The identifier shown before the `dm-<n>` name is the WWN of the LUN presented by the Oracle ZFS Storage Appliance, with a bus identifier prepended. In the example case, the bus identifier is '3'.

The WWN are displayed under the GUID in the browser user interface (BUI) of the Oracle ZFS Storage Appliance shown in Figure 3.

Figure 3. Presented LUNs and their symbolic names

From this figure, you can see that there are two LUNs -- one 4GB and one 8GB LUN -- presented to each Oracle Linux server.

Focusing on the Oracle Linux R6 server `ol-r5`, the two WWNs being presented by the Oracle ZFS Storage Appliance are `600144F0C532BDA00000512A754E0001` and `600144F0C532BDA00000512A75600002`.

This is verified in the output of the `multipath -ll` command previously shown. Note that there are case differences between the iSCSI Qualified Name (IQN) reported by the Oracle ZFS Storage Appliance and the IQN required by DM-Multipath.

In the example, the 4GB volume will be used for Oracle ASM, so it is given the symbolic name `ora_redo`, and the 8GB volume will be used for applications, so it is called `app_space`.

In order to provide these names to the specific LUNs, you must edit the `/etc/multipath.conf` file once more. Adding a `multipaths` declaration will nominate the WWNs of the LUNs.

In the following example, the Oracle volume will be shown.

```
multipaths {
        multipath {
                wwid            3600144f0c532bda00000512a754e0001
                alias           ora_redo
        }

        multipath {
                wwid            3600144f0c532bda00000512a75600002
                alias           app_space
        }
}
```

As with the device declaration, multiple `multipath {…}` declarations can be placed in a `multipaths {…}` statement.

The `/etc/multipath.conf` file will now contain something similar to the following screen output – the WWNs will of course differ according to your particular circumstances.

```
#
# We don't want user_friendly_names as this results in just WWNs being used
# Our configuration has symbolic names for the LUNs to allow automatic
processing
# by UDEV later...
#
defaults {
        user_friendly_names     no
}

devices {
        device  {
                vendor                  "SUN"
                path_checker            readsector0
                hardware_handler        0
                failback                15
                rr_weight               priorities
```

```
                no_path_retry           queue
        }
}

multipaths {
        multipath {
                wwid            3600144f0c532bda00000512a754e0001
                alias           ora_redo
        }

        multipath {
                wwid            3600144f0c532bda00000512a75600002
                alias           app_space
        }
}
```

Finally, DM-Multipath must re-read the configuration to provide the chosen names.

```
[root@ol-r5 ~]# multipath -r
rename: ora_redo (3600144f0c532bda00000512a754e0001) undef SUN,ZFS Storage 7420
size=4.0G features='0' hwhandler='0' wp=undef
|-+- policy='round-robin 0' prio=1 status=undef
| `- 4:0:0:0 sdb 8:16 active ready running
|-+- policy='round-robin 0' prio=1 status=undef
| `- 5:0:0:0 sdc 8:32 active ready running
`-+- policy='round-robin 0' prio=1 status=undef
  `- 3:0:0:0 sdd 8:48 active ready running
Jun 11 17:55:48 | 3600144f0c532bda00000512a75600002: rename
3600144f0c532bda00000512a75600002 to app_space
rename: app_space (3600144f0c532bda00000512a75600002) undef SUN,ZFS Storage 7420
size=8.0G features='0' hwhandler='0' wp=undef
|-+- policy='round-robin 0' prio=1 status=undef
| `- 4:0:0:1 sde 8:64 active ready running
|-+- policy='round-robin 0' prio=1 status=undef
| `- 5:0:0:1 sdf 8:80 active ready running
`-+- policy='round-robin 0' prio=1 status=undef
  `- 3:0:0:1 sdg 8:96 active ready running
[root@ol-r5 ~]# cd /dev/mapper
[root@ol-r5 mapper]# ls -l
total 0
brw-rw---- 1 root disk 253,  3 Jun 11 17:51 app_space
crw------- 1 root root  10, 62 Jun 11 17:51 control
brw-rw---- 1 root disk 253,  2 Jun 11 17:51 ora_redo
brw-rw---- 1 root disk 253,  0 Jun 11 17:51 VolGroup00-LogVol00
brw-rw---- 1 root disk 253,  1 Jun 11 17:51 VolGroup00-LogVol01
[root@ol-r5 mapper]#
```

DM-Multipath will now use these names for the device nodes through every reboot rather than apply the non-deterministic `dm-<n>` names, which can differ between reboots, depending on the discovery order.

DM-Multipath Device Nodes and Symbolic Links

DM-Multipath will automatically create and maintain additional links and device nodes for internal and external use. The following table describes the naming and use of these links and device nodes:

DEVICE / SYMBOLIC LINK	USE
`/dev/sd*`	For each LUN accessible by the server, there will be a `/dev/sd` entry for every path to that LUN. These should not be used directly by applications.
`/dev/dm-*`	Internal DM-Multipath names – these should not be used directly by applications.
`/dev/mapper/*` `/dev/mpath/*`	The `/dev/mapper/*` entries are created early in the boot process and are used for volumes required during this process. The `/dev/mpath/*` entries are created later and are meant for use by applications started after the boot process completes. The `/dev/mpath` directory is created as a convenience to hold all `/dev/mpath/*` entries in the same accessible location. These devices (entries) are not available during the boot process.

`/dev/disk/by-id/x`
`/dev/disk/by-label/x`
`/dev/disk/by-path/x`
`/dev/disk/by-uuid/x`
	Symbolic links are created in the `/dev/disk` directories to allow access to storage through different methods. The created link names tend to be too lengthy for use in application configuration.

Verifying DM-Multipath Configuration

From the output of `multipath –ll`, you can see that there are three active paths for each volume:

```
[root@ol-r5 mapper]# multipath -ll
app_space (3600144f0c532bda00000512a75600002) dm-3 SUN,ZFS Storage 7420
size=8.0G features='1 queue_if_no_path' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=1 status=enabled
| `- 4:0:0:1 sde 8:64 active ready running
|-+- policy='round-robin 0' prio=1 status=enabled
| `- 5:0:0:1 sdf 8:80 active ready running
`-+- policy='round-robin 0' prio=1 status=enabled
  `- 3:0:0:1 sdg 8:96 active ready running
ora_redo (3600144f0c532bda00000512a754e0001) dm-2 SUN,ZFS Storage 7420
```

```
size=4.0G features='1 queue_if_no_path' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=1 status=enabled
| `- 4:0:0:0 sdb 8:16 active ready running
|-+- policy='round-robin 0' prio=1 status=enabled
| `- 5:0:0:0 sdc 8:32 active ready running
`-+- policy='round-robin 0' prio=1 status=enabled
  `- 3:0:0:0 sdd 8:48 active ready running
[root@ol-r5 mapper]#
```

In the code example, the paths exist at 3:0:0, 4:0:0 and 5:0:0, and all are active, ready and running.

Verifying Path-failure Detection in DM Multipath

Having multiple paths is only useful if DM-Multipath can operate the existing paths effectively in the case of failure. To test that this is indeed possible, the following example shows a workload being generated on one of the volumes and a failure simulated to ensure that the correct error recovery is carried out and I/O will continue to flow despite the failure.

In one terminal session, the following command is run:

```
[root@ol-r5 ~]# while /bin/true
> do
> dd if=/dev/mapper/app_space of=/dev/null
> done
```

In another terminal session, a continuous tail of the file `/var/log/messages` is performed to watch the state of DM-Multipath during the failure.

```
[root@ol-r5 ~]# tail -f /var/log/messages &
[root@ol-r5 ~]# ifconfig eth3 down
[root@ol-r5 ~]# Jun 11 17:59:19 ol-r5 avahi-daemon[3452]: Interface eth3.IPv6
no longer relevant for mDNS.
Jun 11 17:59:19 ol-r5 avahi-daemon[3452]: Leaving mDNS multicast group on
interface eth3.IPv6 with address fe80::a00:27ff:fe96:2c9c.
Jun 11 17:59:19 ol-r5 avahi-daemon[3452]: Interface eth3.IPv4 no longer
relevant for mDNS.
Jun 11 17:59:19 ol-r5 avahi-daemon[3452]: Leaving mDNS multicast group on
interface eth3.IPv4 with address 192.168.4.82.
Jun 11 17:59:19 ol-r5 avahi-daemon[3452]: Withdrawing address record for
fe80::a00:27ff:fe96:2c9c on eth3.
Jun 11 17:59:19 ol-r5 avahi-daemon[3452]: Withdrawing address record for
192.168.4.82 on eth3.
```

```
Jun 11 17:59:24 ol-r5 kernel:  connection3:0: ping timeout of 5 secs expired,
recv timeout 5, last rx 4295174128, last ping 4295179128, now 4295184128
Jun 11 17:59:24 ol-r5 kernel:  connection3:0: detected conn error (1011)
Jun 11 17:59:25 ol-r5 iscsid: Kernel reported iSCSI connection 3:0 error
(1011) state (3)
Jun 11 17:59:25 ol-r5 iscsid: Kernel reported iSCSI connection 3:0 error
(1011) state (3)
Jun 11 18:01:24 ol-r5 kernel:  session3: session recovery timed out after 120
secs
Jun 11 18:01:24 ol-r5 multipathd: ora_redo: sdc - readsector0 checker reports
path is down
Jun 11 18:01:24 ol-r5 multipathd: checker failed path 8:32 in map ora_redo
Jun 11 18:01:24 ol-r5 kernel: device-mapper: multipath: Failing path 8:32.
Jun 11 18:01:24 ol-r5 kernel: device-mapper: multipath: Failing path 8:80.
Jun 11 18:01:24 ol-r5 multipathd: ora_redo: remaining active paths: 2
Jun 11 18:01:24 ol-r5 multipathd: app_space: sdf - readsector0 checker
reports path is down
Jun 11 18:01:24 ol-r5 multipathd: checker failed path 8:80 in map app_space
Jun 11 18:01:24 ol-r5 multipathd: app_space: remaining active paths: 2
Jun 11 18:01:24 ol-r5 multipathd: dm-3: add map (uevent)
Jun 11 18:01:24 ol-r5 multipathd: dm-3: devmap already registered
Jun 11 18:01:24 ol-r5 multipathd: dm-2: add map (uevent)
Jun 11 18:01:24 ol-r5 multipathd: dm-2: devmap already registered
Jun 11 18:01:26 ol-r5 iscsid: connect to 192.168.4.13:3260 failed (No route
to host)
Jun 11 18:01:29 ol-r5 multipathd: ora_redo: sdc - readsector0 checker reports
path is down
Jun 11 18:01:29 ol-r5 multipathd: app_space: sdf - readsector0 checker
reports path is down
Jun 11 18:01:32 ol-r5 iscsid: connect to 192.168.4.13:3260 failed (No route
to host)
Jun 11 18:01:34 ol-r5 multipathd: ora_redo: sdc - readsector0 checker reports
path is down
Jun 11 18:01:34 ol-r5 multipathd: app_space: sdf - readsector0 checker
reports path is down
Jun 11 18:01:38 ol-r5 iscsid: connect to 192.168.4.13:3260 failed (No route
to host)
Jun 11 18:01:39 ol-r5 multipathd: ora_redo: sdc - readsector0 checker reports
path is down
Jun 11 18:01:39 ol-r5 multipathd: app_space: sdf - readsector0 checker
reports path is down
Jun 11 18:01:44 ol-r5 iscsid: connect to 192.168.4.13:3260 failed (No route
to host)
Jun 11 18:01:44 ol-r5 multipathd: ora_redo: sdc - readsector0 checker reports
path is down
Jun 11 18:01:44 ol-r5 multipathd: app_space: sdf - readsector0 checker
reports path is down
Jun 11 18:01:49 ol-r5 multipathd: ora_redo: sdc - readsector0 checker reports
path is down
Jun 11 18:01:49 ol-r5 multipathd: app_space: sdf - readsector0 checker
reports path is down
```

The section of log file highlighted in green shows the iSCSI subsystem noting that the paths have failed for the appropriate time (as configured in `/etc/iscsi/iscsid.conf`).

DM-Multipath also notes that the number of paths has been reduced from the three configured to only two. It is important to note that this is discovered not only for the volume on which the workload is being imposed but also the idle volume.

DM-Multipath then marks the failed paths as faulted (highlighted in green) in the following **`multipath -ll`** command output.

```
[root@ol-r5 ~]# multipath -ll
app_space (3600144f0c532bda00000512a75600002) dm-3 SUN,ZFS Storage 7420
size=8.0G features='1 queue_if_no_path' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=1 status=active
| `- 4:0:0:1 sde 8:64 active ready  running
|-+- policy='round-robin 0' prio=0 status=enabled
| `- 5:0:0:1 sdf 8:80 failed faulty running
`-+- policy='round-robin 0' prio=1 status=enabled
  `- 3:0:0:1 sdg 8:96 active ready  running
ora_redo (3600144f0c532bda00000512a754e0001) dm-2 SUN,ZFS Storage 7420
size=4.0G features='1 queue_if_no_path' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=1 status=enabled
| `- 4:0:0:0 sdb 8:16 active ready  running
|-+- policy='round-robin 0' prio=0 status=enabled
| `- 5:0:0:0 sdc 8:32 failed faulty running
`-+- policy='round-robin 0' prio=1 status=enabled
  `- 3:0:0:0 sdd 8:48 active ready  running
```

Importantly, DM-Multipath notes failure of the path between the Oracle ZFS Storage Appliance and the Oracle Linux host, and also recovery of the path. As can be seen in the following code example, DM-Multipath automatically re-enables the path when it becomes available once more.

```
[root@ol-r5 ~]# ifconfig eth3 up
Jun 11 18:06:49 ol-r5 kernel: e1000: eth3 NIC Link is Up 1000 Mbps Full
Duplex, Flow Control: RX
Jun 11 18:06:49 ol-r5 avahi-daemon[3452]: New relevant interface eth3.IPv4
for mDNS.
Jun 11 18:06:49 ol-r5 avahi-daemon[3452]: Joining mDNS multicast group on
interface eth3.IPv4 with address 192.168.4.82.
Jun 11 18:06:49 ol-r5 avahi-daemon[3452]: Registering new address record for
192.168.4.82 on eth3.
```

```
Jun 11 18:06:50 ol-r5 avahi-daemon[3452]: New relevant interface eth3.IPv6
for mDNS.
Jun 11 18:06:50 ol-r5 avahi-daemon[3452]: Joining mDNS multicast group on
interface eth3.IPv6 with address fe80::a00:27ff:fe96:2c9c.
Jun 11 18:06:50 ol-r5 avahi-daemon[3452]: Registering new address record for
fe80::a00:27ff:fe96:2c9c on eth3.
Jun 11 18:06:50 ol-r5 multipathd: ora_redo: sdc - readsector0 checker reports
path is down
Jun 11 18:06:50 ol-r5 multipathd: app_space: sdf - readsector0 checker
reports path is down
Jun 11 18:06:52 ol-r5 iscsid: connection3:0 is operational after recovery (32
attempts)
Jun 11 18:06:55 ol-r5 multipathd: ora_redo: sdc - readsector0 checker reports
path is up
Jun 11 18:06:55 ol-r5 multipathd: 8:32: reinstated
Jun 11 18:06:55 ol-r5 multipathd: ora_redo: remaining active paths: 3
Jun 11 18:06:55 ol-r5 multipathd: app_space: sdf - readsector0 checker
reports path is up
Jun 11 18:06:55 ol-r5 multipathd: 8:80: reinstated
Jun 11 18:06:55 ol-r5 multipathd: app_space: remaining active paths: 3
Jun 11 18:06:55 ol-r5 multipathd: dm-2: add map (uevent)
Jun 11 18:06:55 ol-r5 multipathd: dm-2: devmap already registered
Jun 11 18:06:55 ol-r5 multipathd: dm-3: add map (uevent)
Jun 11 18:06:55 ol-r5 multipathd: dm-3: devmap already registered
[root@ol-r5 ~]# multipath -ll
app_space (3600144f0c532bda00000512a75600002) dm-3 SUN,ZFS Storage 7420
size=8.0G features='1 queue_if_no_path' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=1 status=active
| `- 4:0:0:1 sde 8:64 active ready running
|-+- policy='round-robin 0' prio=1 status=enabled
| `- 5:0:0:1 sdf 8:80 active ready running
`-+- policy='round-robin 0' prio=1 status=enabled
  `- 3:0:0:1 sdg 8:96 active ready running
ora_redo (3600144f0c532bda00000512a754e0001) dm-2 SUN,ZFS Storage 7420
size=4.0G features='1 queue_if_no_path' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=1 status=enabled
| `- 4:0:0:0 sdb 8:16 active ready running
|-+- policy='round-robin 0' prio=1 status=enabled
| `- 5:0:0:0 sdc 8:32 active ready running
`-+- policy='round-robin 0' prio=1 status=enabled
  `- 3:0:0:0 sdd 8:48 active ready running
```

When the path failure occurs on the Oracle ZFS Storage Appliance side, the error messages are slightly different, as the Oracle Linux server does not report errors for Avahi networking, for instance. However, this can be a valuable clue in fault tracing. The following shows the error messages presented on the Oracle Linux server when the failure occurs in this manner. The failure is noted in the yellow highlighted area and resolution of the problem notified in the green highlighted lines.

```
Jun 11 18:11:15 ol-r5 kernel:  connection1:0: ping timeout of 5 secs expired,
recv timeout 5, last rx 4295885428, last ping 4295890428, now 4295895428
Jun 11 18:11:15 ol-r5 kernel:  connection1:0: detected conn error (1011)
Jun 11 18:11:15 ol-r5 iscsid: Kernel reported iSCSI connection 1:0 error
(1011) state (3)
Jun 11 18:11:39 ol-r5 iscsid: connect to 192.168.3.13:3260 failed (No route
to host)
Jun 11 18:12:16 ol-r5 last message repeated 6 times
[root@ol-r5 ~]#  Jun 11 18:13:11 ol-r5 last message repeated 9 times
Jun 11 18:13:15 ol-r5 kernel:  session1: session recovery timed out after 120
secs
Jun 11 18:13:16 ol-r5 multipathd: ora_redo: sdd - readsector0 checker reports
path is down
Jun 11 18:13:16 ol-r5 multipathd: checker failed path 8:48 in map ora_redo
Jun 11 18:13:16 ol-r5 multipathd: ora_redo: remaining active paths: 2
Jun 11 18:13:16 ol-r5 kernel: device-mapper: multipath: Failing path 8:48.
Jun 11 18:13:16 ol-r5 kernel: device-mapper: multipath: Failing path 8:96.
Jun 11 18:13:16 ol-r5 multipathd: app_space: sdg - readsector0 checker
reports path is down
Jun 11 18:13:16 ol-r5 multipathd: checker failed path 8:96 in map app_space
Jun 11 18:13:16 ol-r5 multipathd: app_space: remaining active paths: 2
Jun 11 18:13:17 ol-r5 iscsid: connect to 192.168.3.13:3260 failed (No route
to host)
Jun 11 18:13:17 ol-r5 multipathd: dm-2: add map (uevent)
Jun 11 18:13:17 ol-r5 multipathd: dm-2: devmap already registered
Jun 11 18:13:17 ol-r5 multipathd: dm-3: add map (uevent)
Jun 11 18:13:17 ol-r5 multipathd: dm-3: devmap already registered
Jun 11 18:13:21 ol-r5 multipathd: ora_redo: sdd - readsector0 checker reports
path is down
Jun 11 18:13:21 ol-r5 multipathd: app_space: sdg - readsector0 checker
reports path is down
Jun 11 18:13:23 ol-r5 iscsid: connect to 192.168.3.13:3260 failed (No route
to host)
Jun 11 18:13:26 ol-r5 multipathd: ora_redo: sdd - readsector0 checker reports
path is down
Jun 11 18:13:26 ol-r5 multipathd: app_space: sdg - readsector0 checker
reports path is down
Jun 11 18:13:29 ol-r5 iscsid: connect to 192.168.3.13:3260 failed (No route
to host)
Jun 11 18:13:31 ol-r5 multipathd: ora_redo: sdd - readsector0 checker reports
path is down
Jun 11 18:13:31 ol-r5 multipathd: app_space: sdg - readsector0 checker
reports path is down
Jun 11 18:13:35 ol-r5 iscsid: connect to 192.168.3.13:3260 failed (No route
to host)
Jun 11 18:13:36 ol-r5 multipathd: ora_redo: sdd - readsector0 checker reports
path is down
```

```
Jun 11 18:13:36 ol-r5 multipathd: app_space: sdg - readsector0 checker
reports path is down
Jun 11 18:13:41 ol-r5 iscsid: connect to 192.168.3.13:3260 failed (No route
to host)
Jun 11 18:13:41 ol-r5 multipathd: ora_redo: sdd - readsector0 checker reports
path is down
Jun 11 18:13:41 ol-r5 multipathd: app_space: sdg - readsector0 checker
reports path is down
Jun 11 18:13:45 ol-r5 iscsid: connection1:0 is operational after recovery (23
attempts)
Jun 11 18:13:46 ol-r5 multipathd: ora_redo: sdd - readsector0 checker reports
path is up
Jun 11 18:13:46 ol-r5 multipathd: 8:48: reinstated
Jun 11 18:13:46 ol-r5 multipathd: ora_redo: remaining active paths: 3
Jun 11 18:13:46 ol-r5 multipathd: app_space: sdg - readsector0 checker
reports path is up
Jun 11 18:13:46 ol-r5 multipathd: 8:96: reinstated
Jun 11 18:13:46 ol-r5 multipathd: app_space: remaining active paths: 3
Jun 11 18:13:47 ol-r5 multipathd: dm-2: add map (uevent)
Jun 11 18:13:47 ol-r5 multipathd: dm-2: devmap already registered
Jun 11 18:13:47 ol-r5 multipathd: dm-3: add map (uevent)
Jun 11 18:13:47 ol-r5 multipathd: dm-3: devmap already registered
```

Device Ownership

In certain circumstances, it is important that ownership of volumes is given to a user other than the default – for example, when an application such as a database or a user-space file system requires access to a raw block of storage without the need for a file system structure.

In the case of Oracle systems, these two examples merge together in the form of Oracle Automatic Storage Management (ASM), which takes block storage and a scalable, clusterable file-system-like structure and marries them together in a form that is efficient and manageable within the Oracle environment for the Oracle environment.

Oracle ASM requires that ownership of the device nodes representing the block storage be assigned to a non-root user. Traditionally, this will be the “oracle” user. There are also some useful recommendations regarding the group of the device nodes.

Oracle Linux Release 5

In Oracle Linux Release 5, DM-Multipath has the facility to change ownership and protection modes of devices it creates. By specifying the ‘uid’, ‘gid’ or ‘mode’ of the file, devices can be easily created to allow non-root users to access the storage where appropriate. Symbolic user and group names or integers can be used for the `uid` and `gid` fields and the standard octal representation of permissions for the `mode` field.

The `/etc/multipath.conf` configuration file would be modified similarly to the following:

```
multipaths {
        multipath {
               wwid           3600144f0c532bda00000512a754e0001
               alias          ora_redo
             uid          oracle
             gid          orainst
             mode         660
        }

        multipath {
                wwid            3600144f0c532bda00000512a75600002
                alias           app_space
        }
}
```

Oracle Linux Release 6

Unfortunately, the simple method available under Oracle Linux Release 5 for specifying ownership in the DM-Multipath configuration file is not available under Oracle Linux Release 6.

One possible solution under Release 6 is to use the UDEV facility, which is effectively the device manager in Oracle Linux. It is the successor to the Linux *hotplug* and *devfs* but operates in the user space (which is the reason for the 'U' in the name).

UDEV consists of a daemon user process that is notified of any hardware events requiring its service. The daemon then processes a number of rules that are used to create the necessary symbolic representations of the hardware event. An example of this could be the insertion of a USB flash drive, which will cause block storage device nodes to be created automatically and flash drive contents in a mounted file system presented.

The rules are powerful and have a strict syntax. A number of helpful websites exist whose main topic is the writing and debugging of UDEV rules.

One simple way of determining ownership is to base the user who has access to the LUN on a prefix or some other naming convention.

The examples shown so far have had the volumes `ora_redo` and `app_space`. As mentioned previously, Oracle ASM requires that the LUNs used to host ASM structures are writable by a user other than root. In the following case, the user will be `oracle` and the device nodes should be writable by that user and also by all members of the group `orainst`.

A UDEV rules file is created that changes the ownership of the appropriate devices when they are created or changed – for example, at boot time.

The file `99-diskownership.rules` is created in `/etc/udev/rules.d` and contains the following UDEV rules:

```
SUBSYSTEM!="block",  GOTO="quickexit"
KERNEL!="dm-[0-9]*", GOTO="quickexit"
PROGRAM=="/sbin/dmsetup info -c --noheadings -o name -m %m -j %M"
RESULT=="ora_*", OWNER="oracle", GROUP="dba", MODE="0660"
RESULT=="app_*", OWNER="geosurvey", GROUP="geology", MODE="0660"
LABEL="quickexit
```

UDEV processes a large array of different device types, so it is important to ensure that this script is eliminated from running for all but those that are necessary. To filter out events, the script checks that it is not attempting to process any kernel subsystem other than the block storage one and, if it is, it immediately jumps to the end of the rules file and exits.

Next the script checks if the name of the device being processed starts with anything other than `dm-`*`x`* (where *`x`* is a digit) and if it does, processing immediately jumps to the end of the rule file.

If execution continues to the third line, a block device whose name starts with `dm-`*`x`* is being processed, so it is necessary to find which alias has been assigned in the DM-Multipath configuration file. The output of the program will be either the empty string if no alias has been assigned or the appropriate assigned alias name for this multipathed device.

Given that the naming convention used is to prefix all Oracle ASM volumes with the string `ora_`, you can deduce that any devices matching this criterion will need to be owned by the user '`oracle`', be owned by the group '`dba`' and have the security mode '`rw-rw----`'.

These last three attributes are handled by the UDEV declarations '`OWNER`', '`GROUP`' and '`MODE`' by assigning the appropriate values.

This script checking process can be extended for different applications/ownership requirements by simply prefixing the volume name with an appropriate string.

In the example just shown, any volume aliases starting with the string '`app_`' are automatically changed to ownership of the user '`geosurvey`' and the group '`geology`'. This could be for a geological analysis package that requires access to block storage.

Conclusion

Oracle Linux and the Oracle ZFS Storage Appliance are engineered to perform well together in high-availability situations, allowing redundant connections to storage to be used efficiently and effectively.

Data access continuity and consistent volume naming provide a highly stable platform on which to build business applications, assuring the Oracle ZFS Storage Appliance continuously meets business needs in a changing environment.

References

Oracle ZFS Storage Appliance Documentation	http://www.oracle.com/technetwork/server-storage/sun-unified-storage/documentation/index.html
Oracle Linux Product Pages	http://www.oracle.com/us/technologies/linux/overview/index.html
Oracle ZFS Storage Appliance Product Pages	http://www.oracle.com/us/products/servers-storage/storage/nas/overview/index.html

Configuring Multipathing for Oracle Linux and the Oracle ZFS Storage Appliance

October 2013, Version 1.0
Author: Andrew Ness

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

Oracle is committed to developing practices and products that help protect the environment

Hardware and Software, Engineered to Work Together