ORACLE RMAN ON FLASHBLADE

A Framework for Accelerating Oracle® Backup & Recovery Using FlashBlade™

June 2017

TABLE OF CONTENTS

INTRODUCTION

A robust backup strategy is key to any enterprise Oracle® deployment, and among Oracle database administrators, Oracle RMAN is the most trusted backup and restore utility. RMAN offers a robust and flexible framework for Oracle database backup and recovery. The software includes extensive features – like the ability to detect corruption, centralized cataloging, parallelized backup and recovery operations, and incremental backups. While RMAN may be the most flexible software available, the key challenge that every Oracle database administrator faces is the time it takes to back up and restore a database to and from secondary media. Meeting SLAs is becoming increasingly difficult, especially due to ever-growing database sizes. In addition, today's enterprise demands stringent RPO and RTO to meet regulatory requirements.

To meet with these requirements, storage and backup vendors have come up with purpose-built appliances to speed up the process. They leverage various technologies, like de-duplication, to optimize space requirements on spinning media and accelerate metadata using flash memory storage. While these purpose-built appliances may have optimized backup times, they take a big hit during database restoration. The data rehydration process during recovery operations generates large random I/O access patterns on hard drives, resulting in poor performance. Thus, it becomes very important to understand how quickly a purpose-built appliance can cope up with a disaster when almost every application involved in the ecosystem needs recovery at the same time.

Data is the new currency of our modern cloud era. The speed at which an enterprise can restore data in the event of a disaster defines business edge – and also defines a new class of solution area, called "Fast Backup/Restore".

THE SOLUTION

With the introduction of FlashBlade™, a ground-breaking scale-out flash storage system from Pure Storage®, the backup and recovery challenges of enterprise customers can be addressed easily. Organizations can store more recent database backups on FlashBlade, and this data can be restored quickly in the event of a failure. While the performance of FlashBlade is very impressive, even more attractive its non-disruptive scalability. FlashBlade capacity and performance can be scaled completely non-disruptively, simplifying operational procedures in an organization's backup/recovery process. The system can be scaled in seconds, without running special commands or connecting extra cables. Additionally, the in-line compression feature of FlashBlade reduces capacity requirements by almost 66% for most typical databases, driving costs down significantly without sacrificing performance. Allowing the storage system to handle compression also significantly saves host CPU cycles. The current FlashBlade system can support a read rate of 16GBps and a write rate of 4.5GBps in a single 4U chassis. The sustained write rate of 4.5GBps is equivalent to a 15 TB/hour backup rate, which is well above the normal required backup rate of several customers. Moreover, the restore rates from FlashBlade can be higher than 45 TB/hour with concurrent restore operations. Customers can realize these backup/restore rates without needing any exotic and costly software on their hosts for block differencing purposes.

OVERVIEW

The purpose of this technical white paper is to provide guidelines and techniques to improve Oracle backup and recovery performance using FlashBlade as the target. The goal of this document is not only to help Oracle database administrators meet their backup/recovery SLAs, but to design a system that will scale with their needs non-disruptively and easily.

AUDIENCE

The target audience for this document includes, but is not limited to: Oracle database administrators, storage administrators, IT managers, system architects, sales engineers, field consultants, professional services, and partners who are looking to design and deploy Oracle RMAN backups and recovery on FlashBlade. A working knowledge of Oracle, Linux®, server, storage, and networks is assumed but is not a prerequisite to reading this document.

ORACLE BACKUP AND RECOVERY

Oracle Recovery Manager (RMAN) is a utility that enables effective backup and recovery of Oracle databases. RMAN utility is available free of cost and comes with Oracle software installation.

The default data repository for RMAN, which contains backup and recovery metadata, is the controlfile of the Oracle database that is being backed up. Archive and backup information are stored in the reusable section of the controlfile. Oracle also provides the option of saving RMAN metadata in an RMAN catalog database, which is the preferred approach when dealing with hundreds of databases with varying retention period policies.

Some of the unique features of RMAN that Oracle DBAs depend on are:

- Ability to detect corrupt blocks in data files
- Ability to backup a database and its components without putting the database in 'hot backup' mode, as required in user-managed backups
- Central repository catalog to track backups across all Oracle databases within an organization
- Cross-platform data conversion (useful for migration)
- Flexible recovery options at the database, tablespace, data file, table, or block level
- Parallelize backup and recovery using multiple processes
- Validation and testing of backups and restores without performing the activity

ORACLE BACKUPS OVERVIEW

Oracle backups can be performed via two methods, Recovery Manager (RMAN) and User-managed backup. RMAN is the preferred mechanism for backups whether using Oracle ASM or host filesystems, as well as for situations requiring logical recovery, like datafile or tablespace recovery. User-managed backups, where RMAN is not used, involve datafiles, control files, and archived logs backed up using Operating System commands or third-party storage snapshot features.

RMAN performs backups of the following file types.

- Data files and control files
- Server parameter file
- Archived redo logs
- RMAN backups

ORACLE dNFS

Oracle dNFS (direct Network File System) is the NFS client functionality directly integrated in the Oracle RDBMS server. dNFS makes the task of configuring an Oracle database on NAS storage much simpler, instead of Standard NFS (aka Kernel NFS). Direct NFS Client on Oracle 12c supports NFSv3, NFSv4, and NFSv4.1 protocols to access the NFS server.

The key benefits of Direct NFS Client include simplicity, ease of administration, load balancing, high availability, and cost effectiveness. Oracle has optimized the I/O code path by avoiding kernel overhead, and as such dNFS can improve I/O performance.

Direct NFS Client is capable of performing concurrent direct I/O by bypassing Operating System level caches. It also performs asynchronous I/O, which allows processing to continue while the I/O request is submitted and processed. These two key performance and scalability features provide unparalleled performance when compared to Kernel NFS clients. Another key feature of Direct NFS Client is high availability. Direct NFS Client delivers optimized performance by automatically load balancing requests across all specified paths (up to four parallel network paths). If one network path fails, then Direct NFS Client will reissue I/O commands over any remaining paths, ensuring fault tolerance and high availability.

One of the primary challenges of Kernel NFS administration is inconsistency with configurations across different platforms. Direct NFS Client eliminates this problem by providing a standard NFS client implementation across all platforms supported by the Oracle Database. This also makes NFS a viable option on platforms like Windows which don't natively support NFS.

As NFS is a shared file system, it supports Real Application Cluster (RAC) databases as well as single instance databases. Direct NFS Client recognizes when an instance is part of an RAC and automatically optimizes the mount points for RAC, relieving the administrator of having to manually configure the NFS parameters.

DIRECT NFS CLIENT USAGE

To use Direct NFS Client, the NFS file systems must first be mounted and available over regular NFS mounts. The mount options used in mounting the file systems are not relevant, as Direct NFS Client manages the configuration after installation. Direct NFS Client searches for the mount entries in the following order. It will use the first entry found if duplicate entries exist in the configuration files.

1. $ORACLE_HOME/dbs/oranfstab
2. /etc/oranfstab
3. /etc/mtab

Sample oranfstab file

```
Server: FlashBlade
path: 192.168.201.100
path: 192.168.202.100
path: 192.168.203.100
path: 192.168.204.100
nfs_version: nfsv3
export: /rman1  mount: /m01
```

For multiple paths that are on the same subnet, static path[1] has to be setup on Linux. For all other platforms, you can include the attribute **dontroute** inside the **oranfstab** file.

ENABLING & DISABLING DIRECT NFS CLIENT

To enable the Direct NFS Client, the standard Oracle Disk Manager (ODM) library that supports Direct NFS Client should be used. It can be enabled as follows.

```
cd $ORACLE_HOME/rdbms/lib
make -f ins_rdbms.mk dnfs_on
```

To disable the Direct NFS Client, perform the following.

```
cd $ORACLE_HOME/rdbms/lib
make -f ins_rdbms.mk dnfs_off
```

[1] *See Oracle Support Document "How to Setup Direct NFS client multipaths in same subnet Doc ID # 822481.1"*

SOLUTION DESIGN

LOGICAL ARCHITECTURE

The Oracle database server was built with Oracle Linux 7.2 on the Cisco UCS® B200 M4 blade. The single instance database was setup on a filesystem hosted on a Pure Storage FlashArray//M50. The RMAN target was a 15-blade FlashBlade system, which was connected through Ethernet; the NFS filesystems from FlashBlade were mounted on the database server.

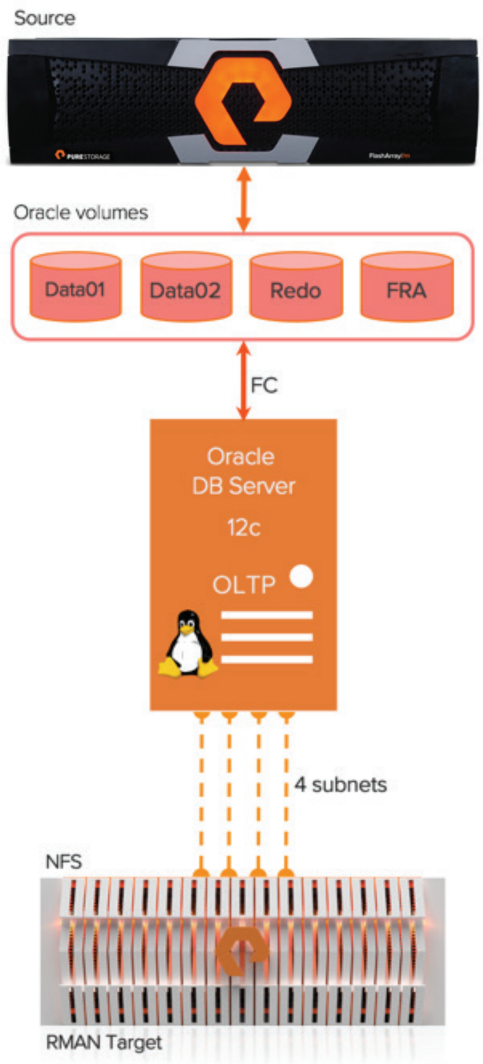


Figure 1. Logical architecture

SERVER CONFIGURATION

An Intel® CPU-based Cisco UCS B-series B200 M4 blade server was deployed to host the single instance Oracle database. The server has Cisco UCS VIC 1340 cards, and they were connected by four ports from each Cisco Fabric extender of the Cisco UCS chassis to the Cisco Fabric Interconnect, which was in turn connected to the Cisco MDS 9148S for upstream connectivity to access FlashArray//M LUNs. The server configuration is described in Table 1.

Component	Description
Processor	2 x Intel Xeon E5-2680 V4 2.4 GHz (2 CPUs with 14 cores each)
Memory	256GB @ 2.4GHz (8 x 32GB)
HBA	4 x 10G Ports on Cisco UCS VIC 1340 (UCSB-MLOM-40G-03) 40Gbps
NIC	6 interfaces configured on Cisco UCS VIC 1340 (1 for Public, 1 for private, 4 for backup)
UCS Firmware (across all components)	3.1 (2b)

Table 1. UCS Blade configuration

The LAN setup under the service profile for the Oracle node was setup with six VLANs and each network interface was configured to one of the VLANs. One interface was used for public traffic, one for private traffic, and the remaining four were for backups. At the host level, the four network interfaces for backups were configured across four subnets on which the NFS filesystems/volumes for the RMAN target from FlashBlade were to be mounted. Four private IP addresses were configured for these four interfaces.

```
[oracle@olxora01 ~]$ ip addr |grep inet |grep 20
    inet 192.168.201.153/24 brd 192.168.201.255 scope global enp8s0
    inet 192.168.202.153/24 brd 192.168.202.255 scope global enp15s0
    inet 192.168.203.153/24 brd 192.168.203.255 scope global enp16s0
    inet 192.168.204.153/24 brd 192.168.204.255 scope global enp17s0

[oracle@olxora01 ~]$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.21.122.1     0.0.0.0         UG    100    0        0 enp6s0
10.21.122.0     0.0.0.0         255.255.255.0   U     100    0        0 enp6s0
192.168.1.0     0.0.0.0         255.255.255.0   U     100    0        0 enp7s0
192.168.201.0   0.0.0.0         255.255.255.0   U     100    0        0 enp8s0
192.168.202.0   0.0.0.0         255.255.255.0   U     100    0        0 enp15s0
```

```
192.168.203.0   0.0.0.0         255.255.255.0   U     100    0        0 enp16s0
192.168.204.0   0.0.0.0         255.255.255.0   U     100    0        0 enp17s0
```

The IP addresses used for the four interfaces for the backup network were 192.168.x.153 for each of the subnets. The four subnets were 201, 202, 203, and 204.

RMAN TARGET – FLASHBLADE CONFIGURATION

Component	Description
FlashBlade	15 x 8TB blades
Capacity	120 TB raw 81.23 TB usable (with no data reduction)
Connectivity	1 x 40GB/s Ethernet (data) 1GB/s Ethernet (Management port)
Physical	4U
Software	Elasticity 2.0.1

Table 2. FlashBlade configuration

FlashBlade network settings were configured with four subnets across four VLANs. The NFS filesystems were to be mounted on these subnets on to the target host.

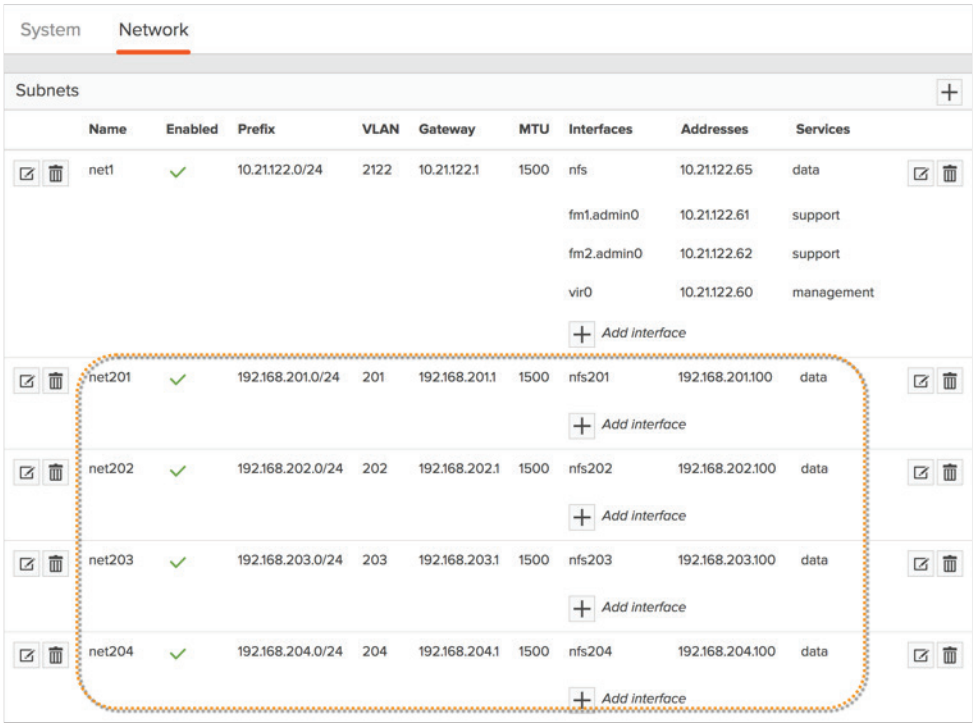


Figure 2. FlashBlade network settings

For the RMAN backup and recovery test, four NFS filesystems were created on FlashBlade, named rman01, rman02, rman03, and rman04, each with a size of 5TB. Each filesystem was mounted onto the host on one of the four subnets.

Note: These filesystems were recreated between tests for easier tracking of space usage details.

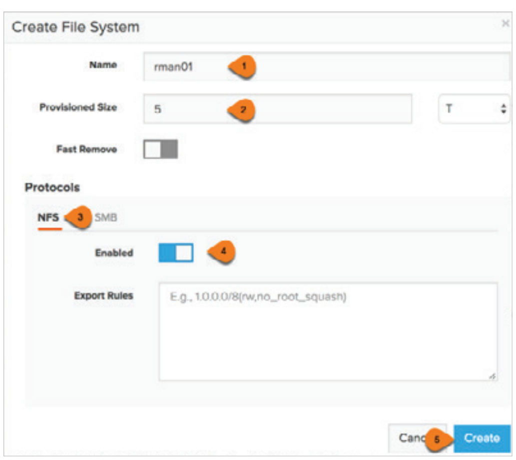


Figure 3. Create file system

Be sure to size the NFS filesystem(s) to hold the RMAN backup. Ideally the filesystems should be sized based on the retention and frequency of full backups. Of course, you will benefit from the data reduction offered by FlashBlade (the rate will vary based on the type of data). As the space is always thin-provisioned, there is no penalty for creating a larger-sized filesystem.

DATABASE CONFIGURATION

An OLTP-type database was setup on Oracle Linux 7.2, and the database was provisioned on the Pure FlashArray//M50. The database was populated with OLTP-type data, and the allocated size of the database was 1.2 TB.

SPACE USAGE OF THE DATABASE

```
Space Details
-------------------------------------------------
Allocated Space :  1,237.27 GB
Used Space      :  1,035.10 GB
```

ORACLE dNFS CONFIGURATION

Oracle dNFS was enabled on the database server based on the test cases (discussed in the **Test Cases & Results** section). For the dNFS test cases, single or multiple NFS filesystems were mounted on the host(s) and the **oranfstab** was updated to reflect the same.

oranfstab file for a single NFS filesystem mounted on a single mount point

```
Server: FlashBlade
path: 192.168.201.100
nfs_version: nfsv3
export: /rman01 mount:/m01
```

oranfstab file for four NFS filesystems mounted on four mount points

```
Server: FlashBlade
path: 192.168.201.100
path: 192.168.202.100
path: 192.168.203.100
path: 192.168.204.100
nfs_version: nfsv3
export: /rman01 mount: /m01
export: /rman02 mount: /m02
export: /rman03 mount: /m03
export: /rman04 mount: /m04
```


TEST CASES AND RESULTS

TEST CASES

To illustrate the advantage of FlashBlade with and without dNFS, the following Oracle RMAN backup and recovery scenarios were tested with FlashBlade as the target.

1. **BACKUP**
   - 1.1. Using Kernel NFS on a single mount point
   - 1.2. Using Kernel NFS with four filesystems mounted on four mount points
   - 1.3. Using dNFS with a single filesystem mounted on a single mount point
   - 1.4. Using dNFS with four filesystems mounted on four mount points
2. **RECOVERY**
   - 2.1. Using Kernel NFS with a single mount point
   - 2.2. Using Kernel NFS with four filesystems mounted on four mount points
   - 2.3. Using dNFS with a single filesystem mounted on a single mount point
   - 2.4. Using dNFS with four filesystems mounted on four mount points

As the environment was setup with four subnets, there was no need to setup static routing or network interface bonding to increase the bandwidth with Kernel NFS testing.

To maximize bandwidth, multiple channels (eight) were used in all test cases.

TEST RESULTS

1.1 BACKUP TO A SINGLE FILESYSTEM ON A SINGLE MOUNT POINT WITH KERNEL NFS

In this test, the dNFS option was disabled at the database level. A single NFS filesystem named **rman01** from FlashBlade was mounted on to the database host using the following command.

```
[root@olxora01 ~]$ mount -t nfs -o rw,bg,nointr,hard,tcp,vers=3,nolock,noac,actim
eo=0 192.168.201.100:/rman01 /m01
```

The disk usage command on the mount point shows the space allocated to the NFS filesystem.

```
[root@olxora01 ~]# df -h /m01
Filesystem              Size  Used Avail Use% Mounted on
192.168.201.100:/rman01 5.0T     0  5.0T   0% /m01
[root@olxora01 ~]#
```

The following command was run on FlashBlade, which shows the NFS connections from the host IP address, which is 192.168.201.153, and on which the NFS filesystem **/rman01** was mounted. The command shows that there is one connection established on NFS port 2049 on blade#10 (ir10).

```
5. ir@SN1-FB-D1-27-2: /opt/ir/admin (ssh)
ir@SN1-FB-D1-27-2:/opt/ir/admin$ ./exec.py -n 1-15 "sudo netstat -atn |grep 2049 |grep ESTA" |grep '192.168.20[1-4].153'
[ir10.out] tcp        0      0 192.168.201.100:2049     192.168.201.153:1010     ESTABLISHED
ir@SN1-FB-D1-27-2:/opt/ir/admin$
```

The above command can only be run by Pure Storage support personnel and not by the customer. To add insight into the connectivity details, the command and the output is included.

The database was backed up using the following RMAN run block commands. Notice the destination: all eight RMAN channels were pointing to the same single NFS mount point/m01.

```
run
{
allocate channel c1 device type disk format '/m01/backup/%U';
allocate channel c2 device type disk format '/m01/backup/%U';
allocate channel c3 device type disk format '/m01/backup/%U';
allocate channel c4 device type disk format '/m01/backup/%U';
allocate channel c5 device type disk format '/m01/backup/%U';
allocate channel c6 device type disk format '/m01/backup/%U';
allocate channel c7 device type disk format '/m01/backup/%U';
allocate channel c8 device type disk format '/m01/backup/%U';
backup database;
}
```

Once the backup was initiated, the connectivity information on FlashBlade was verified again.

```
5. ir@SN1-FB-D1-27-2: /opt/ir/admin (ssh)
ir@SN1-FB-D1-27-2:/opt/ir/admin$ date
Tue May 23 21:08:03 UTC 2017
ir@SN1-FB-D1-27-2:/opt/ir/admin$ ./exec.py -n 1-15 "sudo netstat -atn |grep 2049 |grep ESTA" |grep '192.168.20[1-4].153'
[ir10.out] tcp        0    164 192.168.201.100:2049     192.168.201.153:1010     ESTABLISHED
ir@SN1-FB-D1-27-2:/opt/ir/admin$
```

Even though FlashBlade can have up to 8 x 40GB total connectivity, there is a limit per blade, as each blade within FlashBlade is configured with 2 x 10GB connectivity. Each blade is capable of performing a little over 1GB/s of network traffic with the established connection from the client. This doesn't directly reflect on the writes, as there is overhead with writes into FlashBlade due to its N+2 architecture; the maximum writes a 15-blade FlashBlade system can do, at this time, is 4.5GB/s, but the maximum write bandwidth from a single blade is 10GB/s. Meanwhile, the maximum read rate on a 15-blade FlashBlade system is around 16GB/s.

In this instance, even though eight RMAN channels were started, there is only one connection to one of the blades within FlashBlade, which limits the maximum bandwidth to 10Gbps from the blade to the database host. This should illustrate the disadvantage of Kernel NFS.

Checking network statistics from the database host shows that all the writes to RMAN destinations were going through the interface **enp8s0**, which is the interface where the NFS filesystem was mounted.

The **sar –n DEV** command on the host shows that all network traffic was routed through the interface that is tied to the subnet 192.168.201.x, as the NFS filesystem is mounted on that subnet. Also, the maximum bandwidth we noticed is under 10Gbps.

Average:	IFACE	rxpck/s	txpck/s	rxkB/s	txkB/s	rxcmp/s	txcmp/s	rxmcst/s
Average:	enp17s0	0.48	0.06	0.04	0.00	0.00	0.00	0.16
Average:	enp16s0	0.45	0.03	0.04	0.00	0.00	0.00	0.16
Average:	enp6s0	1.75	1.12	0.13	0.85	0.00	0.00	0.09
Average:	enp7s0	0.06	0.00	0.01	0.00	0.00	0.00	0.04
Average:	lo	0.03	0.03	0.00	0.00	0.00	0.00	0.00
Average:	enp8s0	24119.26	780654.66	2087.69	1155301.68	0.00	0.00	0.16
Average:	enp15s0	0.48	0.06	0.04	0.00	0.00	0.00	0.16

Even though the total allocated space for the database is 1.2TB, the actual used space is 1.01TB, which is exactly the amount that is backed up by RMAN; the rest is all white space. (The actual size was 1086637056, which is equivalent to 1.01TB, but the **df –h** command shows 1.1TB due to formatting.)

```
1. oracle@olxora01:~
[oracle@olxora01 ~]$ df -h /m01
Filesystem                 Size  Used Avail Use% Mounted on
192.168.201.100:/rman01    5.0T  1.1T  4.0T  21% /m01
[oracle@olxora01 ~]$
```

The FlashBlade GUI shows sustained write bandwidth of around **1.05GB/s** for most of the activity.

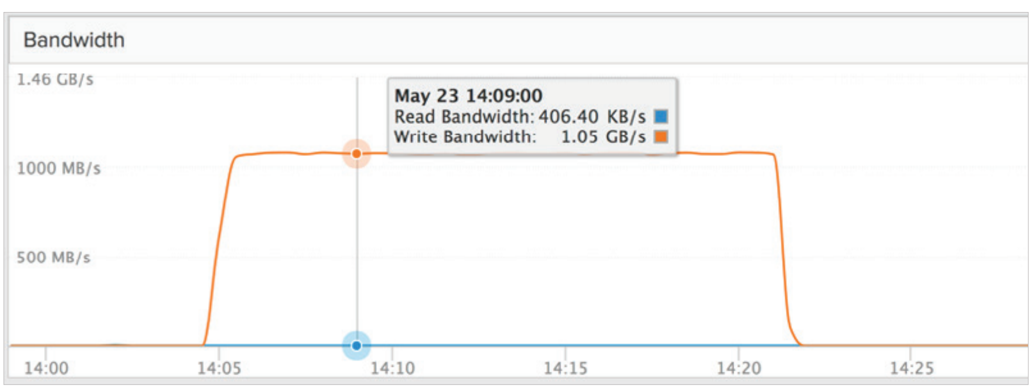


Figure 3. FlashBlade bandwidth

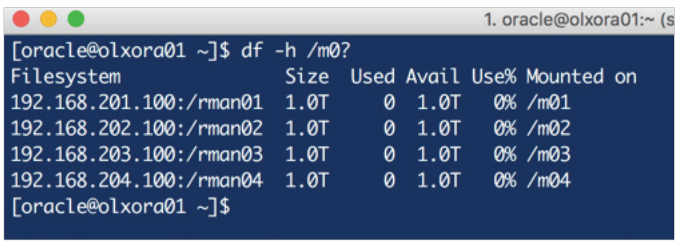
The overall time taken by the backup command as reported in **v$rman_backup_job_details** was 16:31 minutes, which is equivalent to **1.04GB/s** bandwidth.

```
SQL> select start_time, INPUT_BYTES_DISPLAY, OUTPUT_BYTES_DISPLAY, INPUT_BYTES_PER_SEC_DISPLAY,
       OUTPUT_BYTES_PER_SEC_DISPLAY, TIME_TAKEN_DISPLAY
 from v$rman_backup_job_details;

Start time        INPUT_BYTE OUTPUT_BYT In GBps    Out GBps   Elapsed
----------------- ----------- ------------ ---------- ---------- ----------
05/23/17 14:03:39      1.01T      1.01T       1.04G      1.04G      00:16:31
```

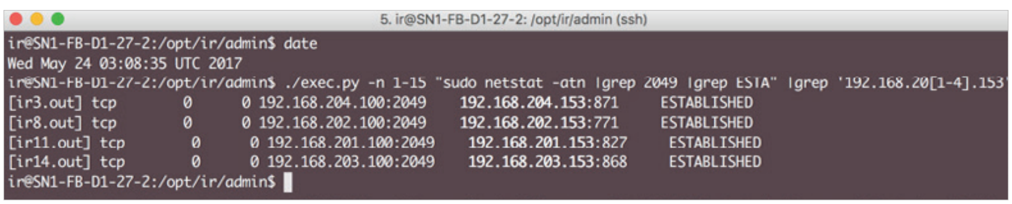
1.2 BACKUP TO FOUR FILESYSTEMS ON FOUR MOUNT POINTS WITH KERNEL NFS

In this test, dNFS was still disabled and four NFS filesystems named rman01 – rman04 from FlashBlade were mounted on to the database host on four different subnets. The following screenshot shows the disk usage of the four filesystems from the host.



```
[oracle@olxora01 ~]$ df -h /m0?
Filesystem               Size  Used Avail Use% Mounted on
192.168.201.100:/rman01  1.0T     0  1.0T   0% /m01
192.168.202.100:/rman02  1.0T     0  1.0T   0% /m02
192.168.203.100:/rman03  1.0T     0  1.0T   0% /m03
192.168.204.100:/rman04  1.0T     0  1.0T   0% /m04
[oracle@olxora01 ~]$
```

Checking the network connections on FlashBlade shows four connections from the host across four subnets on which the four NFS filesystems rman01 to rman04 were mounted.



```
ir@SN1-FB-D1-27-2:/opt/ir/admin$ date
Wed May 24 03:08:35 UTC 2017
ir@SN1-FB-D1-27-2:/opt/ir/admin$ ./exec.py -n 1-15 "sudo netstat -atn |grep 2049 |grep ESTA" |grep '192.168.20[1-4].153'
[ir3.out] tcp        0      0 192.168.204.100:2049    192.168.204.153:871     ESTABLISHED
[ir8.out] tcp        0      0 192.168.202.100:2049    192.168.202.153:771     ESTABLISHED
[ir11.out] tcp        0      0 192.168.201.100:2049    192.168.201.153:827     ESTABLISHED
[ir14.out] tcp        0      0 192.168.203.100:2049    192.168.203.153:868     ESTABLISHED
ir@SN1-FB-D1-27-2:/opt/ir/admin$
```

The database was backed up using the following RMAN run block commands. Notice the destinations for all eight RMAN channels were pointing to the four mount points /m01, /m02, /m03, and /m04.

```
run
{
allocate channel c1 device type disk maxopenfiles 8 format '/m01/orclbkup/%d_%u_s%s_p%p';
allocate channel c2 device type disk maxopenfiles 8 format '/m02/orclbkup/%d_%u_s%s_p%p';
allocate channel c3 device type disk maxopenfiles 8 format '/m03/orclbkup/%d_%u_s%s_p%p';
```

```
allocate channel c4 device type disk maxopenfiles 8 format '/m04/
orclbkup/%d _ %u _ s%s _ p%p';
allocate channel c5 device type disk maxopenfiles 8 format '/m01/
orclbkup/%d _ %u _ s%s _ p%p';
allocate channel c6 device type disk maxopenfiles 8 format '/m02/
orclbkup/%d _ %u _ s%s _ p%p';
allocate channel c7 device type disk maxopenfiles 8 format '/m03/
orclbkup/%d _ %u _ s%s _ p%p';
allocate channel c8 device type disk maxopenfiles 8 format '/m04/
orclbkup/%d _ %u _ s%s _ p%p';
backup database;
}
```

As this test was performed with kernel NFS, the following query against **v$dnfs_servers** shows no entries, as expected.

```
SYS@OLTP>  select svrname, dirname, mntport, nfsport, wtmax, rtmax
  2  from v$dnfs_servers;

no rows selected

SYS@OLTP>
```

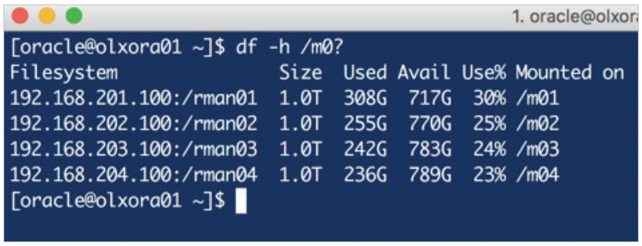
Once the backup was initiated, checking connection information on the backend of FlashBlade shows the same four connections that were connected on four different subnets and as such will be limited to 4 x 10Gb/s of bandwidth.

```
5. ir@SN1-FB-D1-27-2: /opt/ir/admin (ssh)
ir@SN1-FB-D1-27-2:/opt/ir/admin$ date
Wed May 24 03:09:39 UTC 2017
ir@SN1-FB-D1-27-2:/opt/ir/admin$ ./exec.py -n 1-15 "sudo netstat -atn |grep 2049 |grep ESTA" |grep '192.168.20[1-4].153
[ir3.out] tcp        0      0 192.168.204.100:2049    192.168.204.153:871     ESTABLISHED
[ir8.out] tcp    65160      0 192.168.202.100:2049    192.168.202.153:771     ESTABLISHED
[ir11.out] tcp   39096      0 192.168.201.100:2049    192.168.201.153:827     ESTABLISHED
[ir14.out] tcp       0    328 192.168.203.100:2049    192.168.203.153:868     ESTABLISHED
ir@SN1-FB-D1-27-2:/opt/ir/admin$
```

Checking network statistics shows that network traffic from the host was spread across the four backup network interfaces.

08:10:43 PM	IFACE	rxpck/s	txpck/s	rxkB/s	txkB/s	rxcmp/s	txcmp/s	rxmcst/s
08:10:44 PM	enp17s0	22297.00	720820.00	1929.86	1066739.37	0.00	0.00	0.00
08:10:44 PM	enp16s0	30502.00	741748.00	2531.61	1097674.53	0.00	0.00	0.00
08:10:44 PM	enp6s0	2.00	1.00	0.13	0.84	0.00	0.00	0.00
08:10:44 PM	enp7s0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
08:10:44 PM	lo	0.00	0.00	0.00	0.00	0.00	0.00	0.00
08:10:44 PM	enp8s0	23128.00	733910.00	2009.08	1086108.55	0.00	0.00	0.00
08:10:44 PM	enp15s0	22881.00	728242.00	1974.94	1077732.15	0.00	0.00	0.00

Once the backup is complete, checking disk usage across four filesystems shows that the backup of 1.01TB was spread across all four filesystems.



The bandwidth metrics from the FlashBlade GUI show a steady sustained write rate around **3.9GB/s** for most of the backup activity.

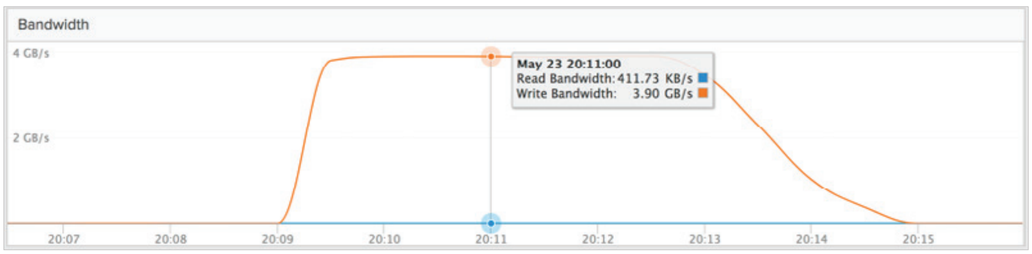


Figure 4. FlashBlade bandwidth

Similarly, the Pure FlashArray which is feeding FlashBlade shows a similar read bandwidth.

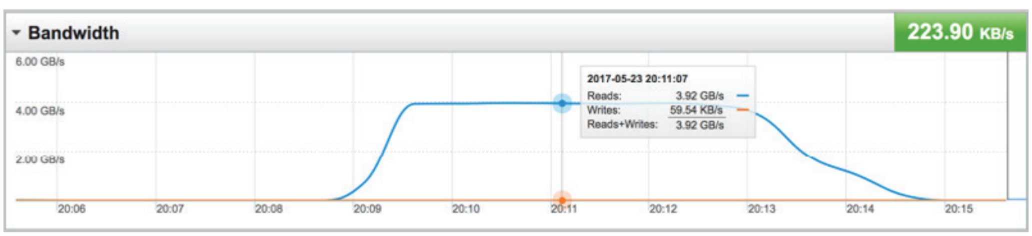


Figure 5. FlashArray bandwidth

The overall time taken by the backup command as reported in **v$rman_backup_job_details** was 5:18 minutes, which is equivalent to **3.25GB/s** of write throughput for 1.01TB of data to be backed up.

Start time	INPUT _ BYTE	OUTPUT _ BYT	In GBps	Out GBps	Elapsed
05/23/17 20:07:56	1.01T	1.01T	3.26G	3.24G	00:05:18

1.3 BACKUP TO SINGLE FILESYSTEM ON A SINGLE MOUNT POINT WITH dNFS ENABLED

For this test, one filesystem named **rman01** from FlashBlade was mounted on the host and dNFS was enabled at the database level.

Following is an excerpt from the **alert.log** file that shows Direct NFS is running.

```
Oracle instance running with ODM: Oracle Direct NFS ODM Library Version 4.0
```

The disk usage command on the mount point shows the space allocated to the NFS filesystem.

```
[root@olxora01 ~]# df -h /m01
Filesystem                  Size  Used Avail Use% Mounted on
192.168.201.100:/rman01  5.0T     0  5.0T   0% /m01
[root@olxora01 ~]#
```

The **oranfstab** file was updated to reflect the four subnets and a single mount point.

```
1. or
[oracle@olxora01 ~]$ more $ORACLE_HOME/dbs/oranfstab
server: FlashBlade
path:192.168.201.100
path:192.168.202.100
path:192.168.203.100
path:192.168.204.100
nfs_version:nfsv3
export:/rman01  mount:/m01
mnt_timeout: 10
[oracle@olxora01 ~]$
```

The following command was run on FlashBlade prior to the RMAN backup command being performed, to verify the NFS connections from the host IP address, which is 192.168.201.153, on which the single NFS filesystem rman01 was mounted. The command shows that there was only one connection established on NFS port 2049 on blade #6 prior to the start of the backup.

```
5. ir@SN1-FB-D1-27-2: /opt/ir/admin (ssh)
ir@SN1-FB-D1-27-2:/opt/ir/admin$ date
Wed May 24 00:56:16 UTC 2017
ir@SN1-FB-D1-27-2:/opt/ir/admin$ ./exec.py -n 1-15 "sudo netstat -atn |grep 2049 |grep ESTA" |grep '192.168.20[1-4].153'
[ir5.out] tcp        0      0 192.168.201.100:2049     192.168.201.153:1021     ESTABLISHED
ir@SN1-FB-D1-27-2:/opt/ir/admin$
```

Even though dNFS is enabled, it only mounts the volume/filesystem and opens the files when they are accessed. As we haven't initiated the RMAN command to back up the database, we do not see the connections at the storage level yet.

Similarly, running a query against **v$dnfs_servers** would show no records as the RMAN commands were not issued.

The database was backed up using the following RMAN run block commands. The destinations for all eight RMAN channels were pointing to the same NFS mount point /m01.

```
run
{
allocate channel c1 device type disk format '/m01/orclbkup/%U';
allocate channel c2 device type disk format '/m01/orclbkup/%U';
allocate channel c3 device type disk format '/m01/orclbkup/%U';
allocate channel c4 device type disk format '/m01/orclbkup/%U';
allocate channel c5 device type disk format '/m01/orclbkup/%U';
allocate channel c6 device type disk format '/m01/orclbkup/%U';
allocate channel c7 device type disk format '/m01/orclbkup/%U';
allocate channel c8 device type disk format '/m01/orclbkup/%U';
backup database;
}
```

Once the backup was started, the dNFS usage was verified by running the query against **v$dnfs_servers**, which shows that the NFS filesystem **/rman01** was mounted.

```
SQL> select svrname, dirname, mntport, nfsport, wtmax, rtmax from v$dnfs _ servers;
SVRNAME              DIRNAME                 MNTPORT    NFSPORT     WTMAX     RTMAX
-------------------- -------------------- ---------- ---------- --------- --------
FlashBlade          /rman01                    2049       2049    524288    524288
```

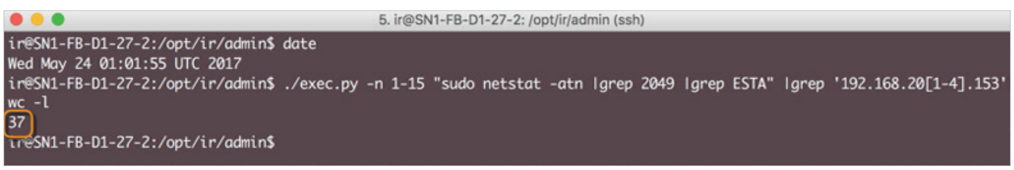
The **v$dnfs_channels** shows that there were 32 channels connected as the backup was run with eight channels and each channel had four paths.

```
SQL> select svrname, path, state, sends, recvs, pings from v$dnfs _ channels;
SVRNAME      PATH            STATE           SENDS      RECVS  PINGS
------------ --------------- ----------- ---------- ---------- ------
FlashBlade   192.168.201.100  CONNECTED       4077       15297      0
FlashBlade   192.168.202.100  CONNECTED       4076       11761      0
FlashBlade   192.168.203.100  CONNECTED       4076       11679      0
FlashBlade   192.168.204.100  CONNECTED       4076       12364      0
FlashBlade   192.168.201.100  CONNECTED       4169       15293      0
FlashBlade   192.168.202.100  CONNECTED       4169       12687      0
FlashBlade   192.168.203.100  CONNECTED       4169       11757      0
FlashBlade   192.168.204.100  CONNECTED       4169       12054      0
FlashBlade   192.168.201.100  CONNECTED       3880       14467      0
FlashBlade   192.168.202.100  CONNECTED       3880       11815      0
FlashBlade   192.168.203.100  CONNECTED       3881       11426      0
FlashBlade   192.168.204.100  CONNECTED       3881       11772      0
FlashBlade   192.168.201.100  CONNECTED       4206       15377      0
FlashBlade   192.168.202.100  CONNECTED       4206       12051      0
```

FlashBlade	192.168.203.100	CONNECTED	4206	11961	0
FlashBlade	192.168.204.100	CONNECTED	4206	12202	0
FlashBlade	192.168.201.100	CONNECTED	4038	15085	0
FlashBlade	192.168.202.100	CONNECTED	4039	12054	0
FlashBlade	192.168.203.100	CONNECTED	4039	11459	0
FlashBlade	192.168.204.100	CONNECTED	4038	11505	0
FlashBlade	192.168.201.100	CONNECTED	4181	15569	0
FlashBlade	192.168.202.100	CONNECTED	4180	11972	0
FlashBlade	192.168.203.100	CONNECTED	4180	12654	0
FlashBlade	192.168.204.100	CONNECTED	4181	12160	0
FlashBlade	192.168.201.100	CONNECTED	4095	15506	0
FlashBlade	192.168.202.100	CONNECTED	4096	11754	0
FlashBlade	192.168.203.100	CONNECTED	4096	11870	0
FlashBlade	192.168.204.100	CONNECTED	4095	11771	0
FlashBlade	192.168.201.100	CONNECTED	3935	15338	0
FlashBlade	192.168.202.100	CONNECTED	3936	11521	0
FlashBlade	192.168.203.100	CONNECTED	3936	11089	0
FlashBlade	192.168.204.100	CONNECTED	3935	11703	0

```
32 rows selected.
```

The connection information was verified at the FlashBlade level again, which shows that there were 37 connections across all blades.


```
5. ir@SN1-FB-D1-27-2: /opt/ir/admin (ssh)
ir@SN1-FB-D1-27-2:/opt/ir/admin$ date
Wed May 24 01:01:55 UTC 2017
ir@SN1-FB-D1-27-2:/opt/ir/admin$ ./exec.py -n 1-15 "sudo netstat -atn |grep 2049 |grep ESTA" |grep '192.168.20[1-4].153'
wc -l
37
ir@SN1-FB-D1-27-2:/opt/ir/admin$
```

Given dNFS is enabled, and with eight RMAN channels, we did see over 32 connections (4 x 8) to different blades in FlashBlade. FlashBlade determines how the connections from clients are established within the blades using various factors, like load. In the case of kernel NFS (when dNFS is disabled), you will only see only one connection per filesystem to FlashBlade, even though RMAN backup could be run with eight channels. This is the major benefit with dNFS: for every server process, it establishes a separate connection to the NFS server for each mount point on each path specified.

The network stats on the host using the **sar –n DEV** command show that the network traffic is spread across all four interfaces.

Average:	IFACE	rxpck/s	txpck/s	rxkB/s	txkB/s	rxcmp/s	txcmp/s	rxmcst/s
Average:	enp17s0	39712.50	904068.99	3274.57	1337612.63	0.00	0.00	0.00
Average:	enp16s0	40782.47	904215.10	3352.52	1337758.33	0.00	0.00	0.00
Average:	enp6s0	2.44	1.30	0.17	0.85	0.00	0.00	0.00
Average:	enp7s0	0.00	0.00	0.00	0.00	0.00	0.00	0.00

```
Average:        lo       0.00       0.00      0.00        0.00      0.00      0.00       0.00
Average:     enp8s0  39941.88  904103.73   3291.82  1337622.45      0.00      0.00       0.00
Average:    enp15s0  40050.65  903781.98   3298.96  1337124.06      0.00      0.00       0.00
```

Bandwidth from the FlashBlade GUI shows a sustained write of over **4.5GB/s** for most of the backup activity.

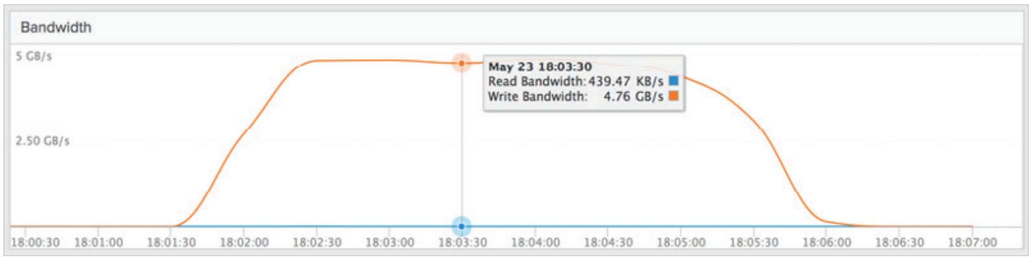


Figure 6. FlashBlade bandwidth

The bandwidth metrics on the source storage system (Pure FlashArray//M50) show similar read bandwidth feeding FlashBlade.

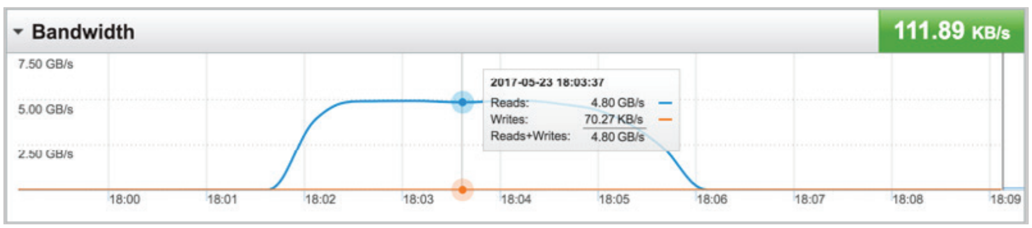


Figure 7. FlashArray bandwidth

The overall time taken by the backup command as reported in v**$rman_backup_job_details** was 3:59 minutes which is equivalent to **4.33GB/s** of throughput and is faster than Kernel NFS with four mount points.

```
SQL> select start_time, INPUT_BYTES_DISPLAY, OUTPUT_BYTES_DISPLAY, INPUT_
BYTES_PER_SEC_DISPLAY, OUTPUT_BYTES_PER_SEC_DISPLAY, TIME_TAKEN_DISPLAY
 from v$rman_backup_job_details;

Start time          INPUT_BYTE OUTPUT_BYT  In GBps    Out GBps    Elapsed
------------------ ------------ ------------ ---------- ---------- ----------
05/23/17 16:26:38    1.01T        1.01T          4.33G      4.31G      00:03:59
```

1.4 BACKUP TO FOUR FILESYSTEMS ON FOUR MOUNT POINTS WITH dNFS ENABLED

For this test, four filesystems/volumes named rman01 to rman04 (sized at 1TB each) from FlashBlade were mounted on to the host.

The disk usage of the filesystems shows each filesystem sized at 1TB.

```
1. oracle@olxora01:~ (s
[oracle@olxora01 ~]$ df -h /m0?
Filesystem             Size  Used Avail Use% Mounted on
192.168.201.100:/rman01  1.0T     0  1.0T   0% /m01
192.168.202.100:/rman02  1.0T     0  1.0T   0% /m02
192.168.203.100:/rman03  1.0T     0  1.0T   0% /m03
192.168.204.100:/rman04  1.0T     0  1.0T   0% /m04
[oracle@olxora01 ~]$
```

The **oranfstab** file was updated to reflect four mount points and four subnets across all eight nodes.

```
1. oracle@olxora
[oracle@olxora01 ~]$ more $ORACLE_HOME/dbs/oranfstab
server: FlashBlade
path:192.168.201.100
path:192.168.202.100
path:192.168.203.100
path:192.168.204.100
nfs_version:nfsv3
export:/rman01  mount:/m01
export:/rman02  mount:/m02
export:/rman03  mount:/m03
export:/rman04  mount:/m04
mnt_timeout: 10
[oracle@olxora01 ~]$
```

Checking the connections on FlashBlade, prior to the RMAN backup command being performed, showed four connections, one for each mount point on four subnets. The connections were established across blades #2, #3, #7, and #12.

```
5. ir@SN1-FB-D1-27-2: /opt/ir/admin (ssh)
ir@SN1-FB-D1-27-2:/opt/ir/admin$ date
Wed May 24 02:18:19 UTC 2017
ir@SN1-FB-D1-27-2:/opt/ir/admin$ ./exec.py -n 1-15 "sudo netstat -atn |grep 2049 |grep ESTA" |grep '192.168.20[1-4].153'
[ir2.out] tcp        0      0 192.168.203.100:2049    192.168.203.153:854     ESTABLISHED
[ir3.out] tcp        0      0 192.168.202.100:2049    192.168.202.153:786     ESTABLISHED
[ir7.out] tcp        0      0 192.168.204.100:2049    192.168.204.153:995     ESTABLISHED
[ir12.out] tcp        0      0 192.168.201.100:2049    192.168.201.153:917     ESTABLISHED
ir@SN1-FB-D1-27-2:/opt/ir/admin$
```

The following RMAN run block command performs the backup of the database with eight RMAN channels, each one pointing to one of the four mount point locations (m01 – m04).

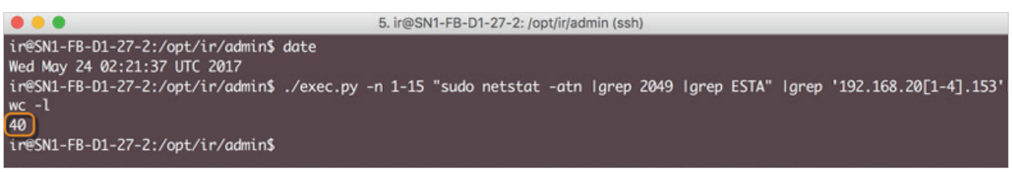
```
run
{
allocate channel c1 device type disk maxopenfiles 8 format '/m01/
orclbkup/%d _ %u _ s%s _ p%p';
allocate channel c2 device type disk maxopenfiles 8 format '/m02/
orclbkup/%d _ %u _ s%s _ p%p';
allocate channel c3 device type disk maxopenfiles 8 format '/m03/
orclbkup/%d _ %u _ s%s _ p%p';
allocate channel c4 device type disk maxopenfiles 8 format '/m04/
orclbkup/%d _ %u _ s%s _ p%p';
```

```
allocate channel c5 device type disk maxopenfiles 8 format '/m01/
orclbkup/%d_%u_s%s_p%p';
allocate channel c6 device type disk maxopenfiles 8 format '/m02/
orclbkup/%d_%u_s%s_p%p';
allocate channel c7 device type disk maxopenfiles 8 format '/m03/
orclbkup/%d_%u_s%s_p%p';
allocate channel c8 device type disk maxopenfiles 8 format '/m04/
orclbkup/%d_%u_s%s_p%p';
backup database;
}
```

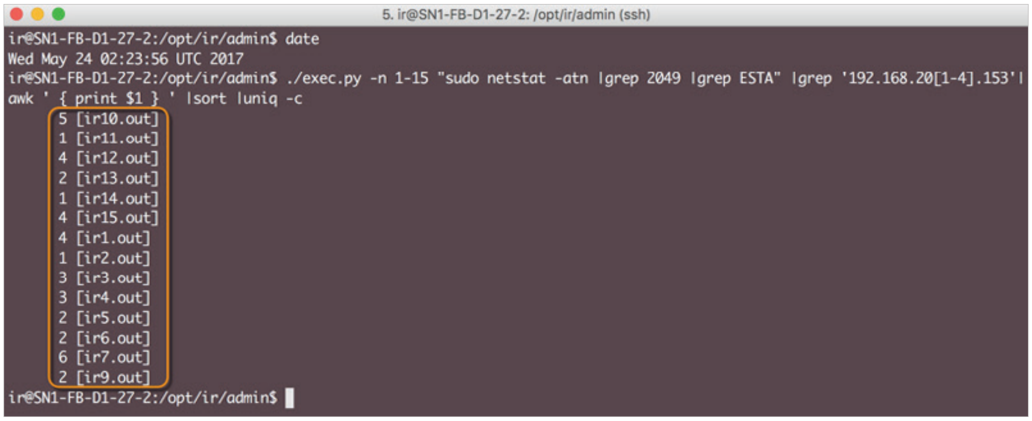
Once the backup was started, the dNFS usage was verified by running the query against **v$dnfs_servers**, which shows the four NFS filesystems rman01 – rman04 were mounted.

```
SQL> select svrname, dirname, mntport, nfsport, wtmax, rtmax from v$dnfs_
servers;
SVRNAME      DIRNAM       MNTPORT    NFSPORT    WTMAX      RTMAX
------------ ----------- ---------- ---------- ---------- ----------
FlashBlade   /rman01           2049       2049     524288     524288
FlashBlade   /rman02           2049       2049     524288     524288
FlashBlade   /rman03           2049       2049     524288     524288
FlashBlade   /rman04           2049       2049     524288     524288
```

As expected, the connection information at the FlashBlade level shows that there were over 32 connections.



Reviewing the connections on FlashBlade shows 40 connections were spread across all 15 blades in FlashBlade.



Checking network traffic using the **sar –n DEV** command shows that they were equally spread across all four backup interfaces, as expected.

Average:	IFACE	rxpck/s	txpck/s	rxkB/s	txkB/s	rxcmp/s	txcmp/s	rxmcst/s
Average:	enp17s0	39712.50	904068.99	3274.57	1337612.63	0.00	0.00	0.00
Average:	enp16s0	40782.47	904215.10	3352.52	1337758.33	0.00	0.00	0.00
Average:	enp6s0	2.44	1.30	0.17	0.85	0.00	0.00	0.00
Average:	enp7s0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Average:	lo	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Average:	enp8s0	39941.88	904103.73	3291.82	1337622.45	0.00	0.00	0.00
Average:	enp15s0	40050.65	903781.98	3298.96	1337124.06	0.00	0.00	0.00

The bandwidth metric on the FlashBlade GUI showed write bandwidth around **4.75GB/s** for most of the backup activity.

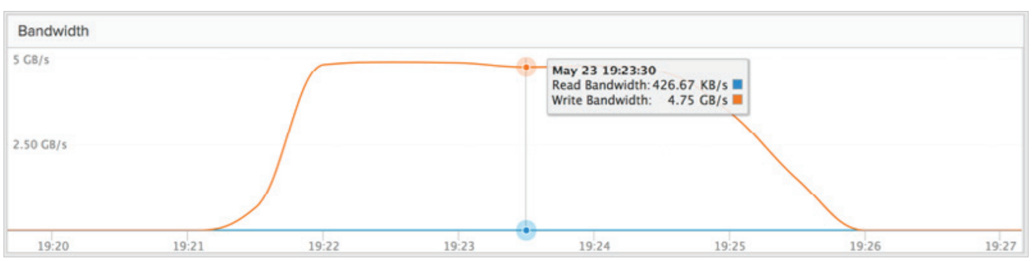


Figure 8. FlashBlade bandwidth

The Pure FlashArray GUI also showed the read bandwidth around **4.8GB/s** for most of the backup activity.

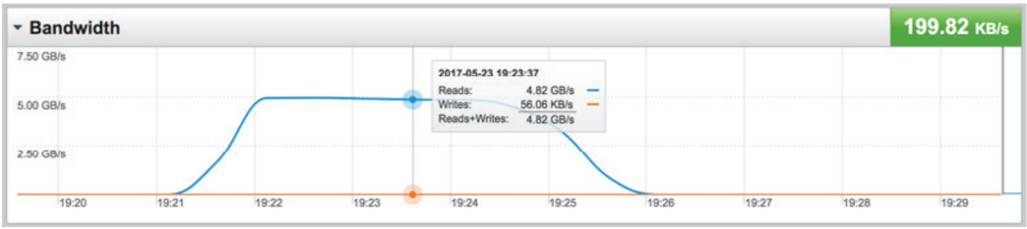


Figure 9. FlashArray bandwidth

Checking disk usage by the filesystem showed that the 1.01TB usage was spread across the four filesystems.

```
1. oracle@olxora01
[oracle@olxora01 ~]$ df -h /m0?
Filesystem                   Size  Used Avail Use% Mounted on
192.168.201.100:/rman01      1.0T  306G  719G  30% /m01
192.168.202.100:/rman02      1.0T  255G  770G  25% /m02
192.168.203.100:/rman03      1.0T  238G  787G  24% /m03
192.168.204.100:/rman04      1.0T  233G  792G  23% /m04
[oracle@olxora01 ~]$
```

The overall time taken by the backup command as reported in **v$rman_backup_job_details** was 03:59 minutes, which was consistent with previous tests. This yields a write bandwidth of **4.33GB/s**.

```
SQL> select start_time, INPUT_BYTES_DISPLAY, OUTPUT_BYTES_DISPLAY, INPUT_BYTES_PER_SEC_DISPLAY, OUTPUT_BYTES_PER_SEC_DISPLAY, TIME_TAKEN_DISPLAY
 from v$rman_backup_job_details;

Start time        INPUT_BYTE OUTPUT_BYT  In GBps    Out GBps    Elapsed
----------------- ------------ ------------ ---------- ---------- ----------
05/23/17 19:20:22        1.01T        1.01T      4.33G      4.31G   00:03:59
```

2.1 RECOVERY FROM A SINGLE FILESYSTEM ON A SINGLE MOUNT POINT WITH KERNEL NFS

This recovery test involves restoring the backup from a single mount point on the database system with Kernel NFS (no dNFS).

Checking FlashBlade for connection information before running the recovery shows a single connection from the 201 subnet on which the filesystem was mounted.

```
5. ir@SN1-FB-D1-27-2: /opt/ir/admin (ssh)
ir@SN1-FB-D1-27-2:/opt/ir/admin$ date
Thu May 25 08:31:30 UTC 2017
ir@SN1-FB-D1-27-2:/opt/ir/admin$ ./exec.py -n 1-15 "sudo netstat -atn |grep 2049 |grep ESTA" |grep '192.168.20[1-4].153'
[ir4.out] tcp        0      0 192.168.201.100:2049     192.168.201.153:719      ESTABLISHED
ir@SN1-FB-D1-27-2:/opt/ir/admin$
```

The following RMAN run block command was used to restore and recover the database.

```
run
{
allocate channel c1 device type disk format '/m01/orclbkup/%U';
allocate channel c2 device type disk format '/m01/orclbkup/%U';
allocate channel c3 device type disk format '/m01/orclbkup/%U';
allocate channel c4 device type disk format '/m01/orclbkup/%U';
allocate channel c5 device type disk format '/m01/orclbkup/%U';
allocate channel c6 device type disk format '/m01/orclbkup/%U';
allocate channel c7 device type disk format '/m01/orclbkup/%U';
allocate channel c8 device type disk format '/m01/orclbkup/%U';
restore database;
recover database;
}
```

Network statistics show that network traffic was routed only through the 201 subnet (Interface **enp8s0**).

```
Average:     IFACE   rxpck/s txpck/s    rxkB/s   txkB/s  rxcmp/s  txcmp/s  rxmcst/s
Average:    enp17s0      0.56     0.00      0.04     0.00     0.00     0.00      0.00
Average:    enp16s0      0.56     0.00      0.04     0.00     0.00     0.00      0.00
Average:     enp6s0      2.67     1.83      0.22     0.90     0.00     0.00      0.00
Average:     enp7s0      0.00     0.00      0.00     0.00     0.00     0.00      0.00
Average:         lo      0.00     0.00      0.00     0.00     0.00     0.00      0.00
Average:     enp8s0 804635.53  48798.46 1194899.93  3604.52     0.00     0.00      0.00
Average:    enp15s0      0.56     0.00      0.04     0.00     0.00     0.00      0.00
```

Once the restore is completed, checking the view **v$session_longops** shows the details of the RMAN full datafile restores, including elapsed time.

```
SYS@OLTP> select opname, start_time, last_update_time, elapsed_seconds,
message from v$session_longops;
OPNAME                        Start Time        Last Upd Time     ELAPSED_SECONDS
----------------------------- ----------------- ----------------- -------------------
MESSAGE --------------------------------------------------------------------------
RMAN: full datafile restore  05/25/17 01:30:55 05/25/17 01:42:54     719
RMAN: full datafile restore: Set Count 473: 15728640 out of 15728640 Blocks done

RMAN: full datafile restore  05/25/17 01:43:03 05/25/17 01:46:43     220
RMAN: full datafile restore: Set Count 474: 15728640 out of 15728640 Blocks done

RMAN: full datafile restore  05/25/17 01:30:57 05/25/17 01:31:12      15
RMAN: full datafile restore: Set Count 477: 952960 out of 952960 Blocks done

RMAN: full datafile restore  05/25/17 01:31:17 05/25/17 01:43:57     760
```

```
RMAN: full datafile restore: Set Count 469: 15728640 out of 15728640 Blocks done

RMAN: full datafile restore  05/25/17 01:30:58 05/25/17 01:43:25     747
RMAN: full datafile restore: Set Count 472: 15728640 out of 15728640 Blocks done

RMAN: full datafile restore  05/25/17 01:30:55 05/25/17 01:42:52     717
RMAN: full datafile restore: Set Count 467: 15728640 out of 15728640 Blocks done

RMAN: full datafile restore  05/25/17 01:31:00 05/25/17 01:43:45     765
RMAN: full datafile restore: Set Count 470: 15728640 out of 15728640 Blocks done

RMAN: full datafile restore  05/25/17 01:31:01 05/25/17 01:43:37     756
RMAN: full datafile restore: Set Count 471: 15728640 out of 15728640 Blocks done

RMAN: aggregate input      05/25/17 01:30:54 05/25/17 01:46:50      956
RMAN: aggregate input: restore 460: 162171520 out of 162171520 Blocks done

RMAN: full datafile restore  05/25/17 01:30:56 05/25/17 01:31:04       8
RMAN: full datafile restore: Set Count 476: 3932160 out of 3932160 Blocks done

RMAN: full datafile restore  05/25/17 01:31:12 05/25/17 01:43:47     755
RMAN: full datafile restore: Set Count 468: 15728640 out of 15728640 Blocks done

RMAN: full datafile restore  05/25/17 01:30:55 05/25/17 01:42:54     719
RMAN: full datafile restore: Set Count 466: 15728640 out of 15728640 Blocks done

RMAN: full datafile restore  05/25/17 01:43:04 05/25/17 01:46:45     221
RMAN: full datafile restore: Set Count 475: 15728640 out of 15728640 Blocks done
```

Based on the above, the overall restore time was 956 seconds to restore 1.01 TB of data at a read bandwidth of **1.08GB/s**.

The FlashBlade GUI shows the bandwidth usage with a maximum around **1.09GB/s** during most of the recovery activity.

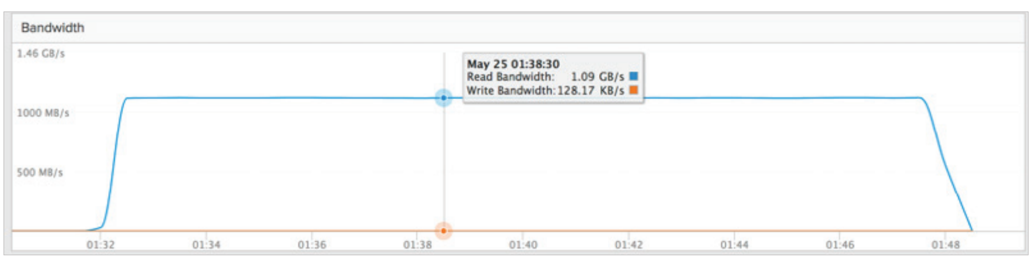


Figure 10. FlashBlade bandwidth

While FlashBlade was used to restore the database, the data was written onto the Pure FlashArray, and its FlashArray GUI shows the write bandwidth averaging **1.09GB/s** for most of the activity, with some peaks over **2GB/s**.

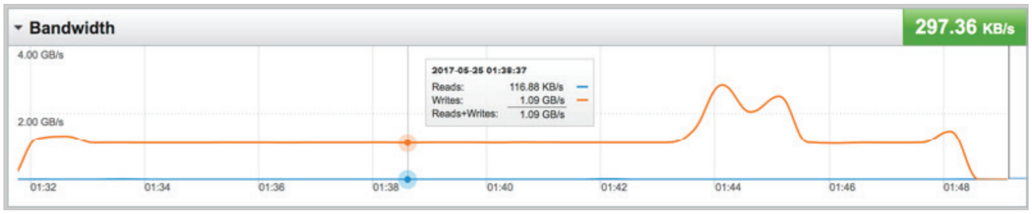


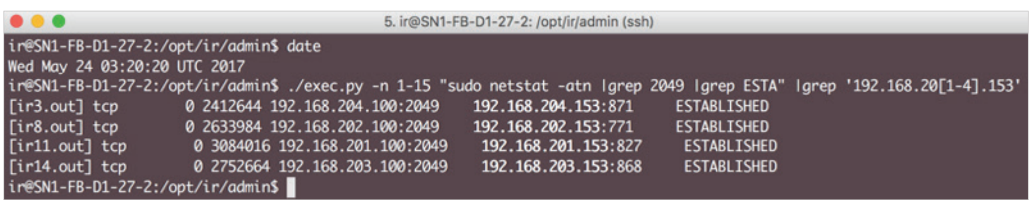
Figure 11. FlashArray bandwidth

2.2 RECOVERY FROM FOUR FILESYSTEMS ON FOUR MOUNT POINTS WITH KERNEL NFS

This recovery test involved restoring a backup from four mount points on the database system with Kernel NFS (no dNFS). The following RMAN run block command was used to restore and recover the database. The files were restored from the four mount points.

```
run
{
allocate channel c1 device type disk format '/m01/orclbkup/%U';
allocate channel c2 device type disk format '/m02/orclbkup/%U';
allocate channel c3 device type disk format '/m03/orclbkup/%U';
allocate channel c4 device type disk format '/m04/orclbkup/%U';
allocate channel c5 device type disk format '/m01/orclbkup/%U';
allocate channel c6 device type disk format '/m02/orclbkup/%U';
allocate channel c7 device type disk format '/m03/orclbkup/%U';
allocate channel c8 device type disk format '/m04/orclbkup/%U';
restore database;
recover database;
}
```

Checking FlashBlade for connection information shows four connections from four subnets.



Network statistics show that network traffic was spread across the four backup network interfaces.

```
Average:      IFACE  rxpck/s   txpck/s    rxkB/s    txkB/s rxcmp/s txcmp/s rxmcst/s
Average:    enp17s0 762068.71  46293.23 1132150.03   3419.41    0.00    0.00     0.00
Average:    enp16s0 538031.77  32216.96  798996.03   2382.23    0.00    0.00     0.00
Average:     enp6s0      2.07      1.43      0.15      0.85    0.00    0.00     0.00
Average:     enp7s0      0.00      0.00      0.00      0.00    0.00    0.00     0.00
Average:         lo      0.16      0.16      0.01      0.01    0.00    0.00     0.00
Average:     enp8s0 803412.26  48842.91 1193179.67   3607.28    0.00    0.00     0.00
Average:    enp15s0 544958.60  32715.13  809289.89   2418.53    0.00    0.00     0.00
```

Once the restore is completed, checking the view **v$session_longops** shows the details of the RMAN full datafile restores, including elapsed time.

```
SYS@OLTP> select opname, start_time, last_update_time, elapsed_seconds,
message from v$session_longops;

OPNAME                       Start Time        Last Upd Time   ELAPSED_SECONDS
---------------------------- ----------------- --------------- ------------------
MESSAGE
--------------------------------------------------------------------------------
RMAN: full datafile restore  05/23/17 20:20:12 05/23/17 20:24:05     233
RMAN: full datafile restore: Set Count 454: 19660800 out of 19660800 Blocks done

RMAN: full datafile restore  05/23/17 20:19:06 05/23/17 20:23:08     242
RMAN: full datafile restore: Set Count 457: 19660800 out of 19660800 Blocks done

RMAN: full datafile restore  05/23/17 20:19:05 05/23/17 20:22:33     208
RMAN: full datafile restore: Set Count 461: 15728640 out of 15728640 Blocks done

RMAN: full datafile restore  05/23/17 20:19:06 05/23/17 20:24:43     337
RMAN: full datafile restore: Set Count 459: 19660800 out of 19660800 Blocks done

RMAN: full datafile restore  05/23/17 20:19:06 05/23/17 20:24:34     328
RMAN: full datafile restore: Set Count 456: 19660800 out of 19660800 Blocks done

RMAN: full datafile restore  05/23/17 20:19:06 05/23/17 20:24:44     338
RMAN: full datafile restore: Set Count 455: 19660800 out of 19660800 Blocks done

RMAN: full datafile restore  05/23/17 20:19:06 05/23/17 20:23:25     259
RMAN: full datafile restore: Set Count 458: 19660800 out of 19660800 Blocks done

RMAN: aggregate input       05/23/17 20:19:05 05/23/17 20:24:49      344
RMAN: aggregate input: restore 460: 162171520 out of 162171520 Blocks done

RMAN: full datafile restore  05/23/17 20:19:05 05/23/17 20:24:06     301
RMAN: full datafile restore: Set Count 460: 16413440 out of 16413440 Blocks done

RMAN: full datafile restore  05/23/17 20:19:05 05/23/17 20:21:23     138
RMAN: full datafile restore: Set Count 462: 8132480 out of 8132480 Blocks done
```

Based on the above, the overall restore time was 344 seconds to restore 1.01 TB of data at a read bandwidth of **3GB/s**. The FlashBlade GUI shows the bandwidth usage with a maximum around **3.58GB/s** during recovery activity.

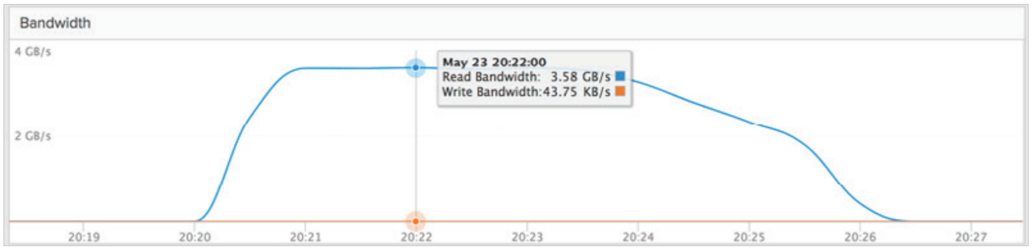


Figure 12. FlashBlade bandwidth

While FlashBlade was used to restore the database, it was written onto Pure FlashArray and its FlashArray GUI shows a write bandwidth averaging **3GB/s** around the same time.

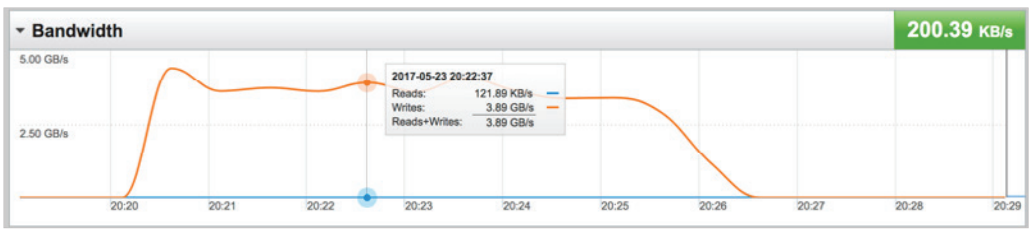


Figure 13. FlashArray bandwidth

2.3 RECOVERY FROM A SINGLE FILESYSTEM ON A SINGLE MOUNT POINT ON dNFS

This recovery test involves restoring a backup from a single mount point (rman01) from the database host with dNFS enabled. The **oranfstab** reflects the four subnets and a single mount point.

```
[oracle@olxora01 ~]$ more $ORACLE_HOME/dbs/oranfstab
server: FlashBlade
path:192.168.201.100
path:192.168.202.100
path:192.168.203.100
path:192.168.204.100
nfs_version:nfsv3
export:/rman01  mount:/m01
mnt_timeout: 10
[oracle@olxora01 ~]$
```

The following RMAN run block command was used to restore and recover the database. The files were restored from the same mount point (/m01) that was mounted on the database host.

```
run
{
allocate channel c1 device type disk format '/m01/orclbkup/%U';
allocate channel c2 device type disk format '/m01/orclbkup/%U';
allocate channel c3 device type disk format '/m01/orclbkup/%U';
allocate channel c4 device type disk format '/m01/orclbkup/%U';
allocate channel c5 device type disk format '/m01/orclbkup/%U';
allocate channel c6 device type disk format '/m01/orclbkup/%U';
allocate channel c7 device type disk format '/m01/orclbkup/%U';
allocate channel c8 device type disk format '/m01/orclbkup/%U';
restore database;
recover database;
}
```

Similar to what we noticed with the backup activity, the restore also enabled multiple connections to FlashBlade from the host. Checking connection information on FlashBlade shows 37 connections spread across various blades.

The **v$session_longops** view shows more information about the restore activity. The earliest start time is 18:26:47, and the last activity time is 18:21:10.

```
SQL> select opname, start_time, last_update_time, elapsed_seconds,  message
from v$session_longops;

OPNAME                        Start Time        Last Upd Time   ELAPSED_SECONDS
---------------------------- ----------------- ---------------- ---------------
MESSAGE
-------------------------------------------------------------------------------
RMAN: full datafile restore  05/23/17 18:15:48 05/23/17 18:17:46     118
RMAN: full datafile restore: Set Count 438: 8132480 out of 8132480 Blocks done

RMAN: full datafile restore  05/23/17 18:15:48 05/23/17 18:20:09     261
RMAN: full datafile restore: Set Count 434: 19660800 out of 19660800 Blocks done

RMAN: full datafile restore  05/23/17 18:15:48 05/23/17 18:20:28     280
```

```
RMAN: full datafile restore: Set Count 430: 19660800 out of 19660800 Blocks done

RMAN: full datafile restore  05/23/17 18:15:48 05/23/17 18:19:35     227
RMAN: full datafile restore: Set Count 437: 15728640 out of 15728640 Blocks done

RMAN: full datafile restore  05/23/17 18:15:48 05/23/17 18:20:32     284
RMAN: full datafile restore: Set Count 435: 19660800 out of 19660800 Blocks done

RMAN: full datafile restore  05/23/17 18:15:48 05/23/17 18:20:37     289
RMAN: full datafile restore: Set Count 433: 19660800 out of 19660800 Blocks done

RMAN: full datafile restore  05/23/17 18:15:48 05/23/17 18:20:36     288
RMAN: full datafile restore: Set Count 431: 19660800 out of 19660800 Blocks done

RMAN: aggregate input        05/23/17 18:15:47 05/23/17 18:21:10     323
RMAN: aggregate input: restore 460: 162171520 out of 162171520 Blocks done

RMAN: full datafile restore  05/23/17 18:15:48 05/23/17 18:19:57     249
RMAN: full datafile restore: Set Count 436: 16413440 out of 16413440 Blocks done

RMAN: full datafile restore  05/23/17 18:16:43 05/23/17 18:20:59     256
RMAN: full datafile restore: Set Count 432: 19660800 out of 19660800 Blocks done
```

Overall the restore of 1.01 TB took 323 seconds which yields an average restore rate of **3.2GB/s** – better than that of the Kernel NFS, especially when using a single mount point.

The FlashBlade GUI shows a read bandwidth of around **4GB/s** for most of the restore activity. Even though FlashBlade is capable of higher bandwidth (in the range of 16GB/s), the limitation is on the receiving end, which is the Pure FlashArray where the database is restored. The FlashArray GUI shows a write bandwidth in the range of **4GB/s** around the same time.

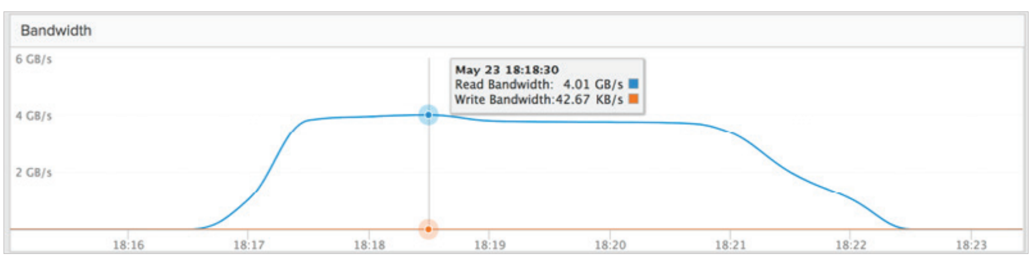


Figure 12. FlashBlade bandwidth

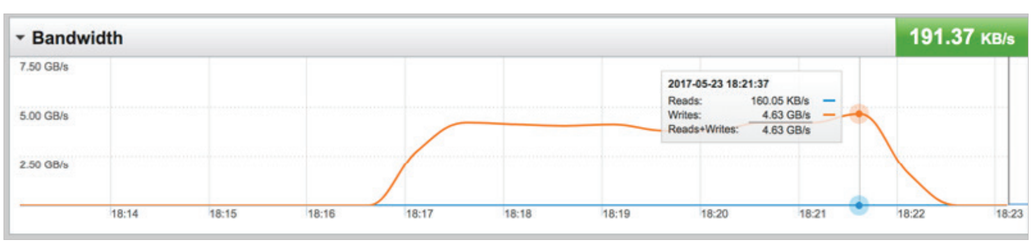


Figure 13. FlashArray bandwidth

2.4 RECOVERY FROM FOUR FILESYSTEMS ON FOUR MOUNT POINTS WITH dNFS ENABLED

This recovery test involves restoring a backup from four filesystems (rman01 – rman04) from the database host with dNFS enabled. As the test was performed immediately after the backup test, there was no change in the **oranfstab**.

```
1. oracle@olxor
[oracle@olxora01 ~]$ more $ORACLE_HOME/dbs/oranfstab
server: FlashBlade
path:192.168.201.100
path:192.168.202.100
path:192.168.203.100
path:192.168.204.100
nfs_version:nfsv3
export:/rman01  mount:/m01
export:/rman02  mount:/m02
export:/rman03  mount:/m03
export:/rman04  mount:/m04
mnt_timeout: 10
[oracle@olxora01 ~]$
```

Prior to performing the restore, checking connection information on FlashBlade shows four connections from the host where the filesystems were mounted on one of the four subnets.

```
ir@SN1-FB-D1-27-2:/opt/ir/admin$ date
Wed May 24 02:50:30 UTC 2017
ir@SN1-FB-D1-27-2:/opt/ir/admin$ ./exec.py -n 1-15 "sudo netstat -atn |grep 2049 |grep ESTA" |grep '192.168.20[1-4].153'
[ir5.out] tcp        0      0 192.168.202.100:2049      192.168.202.153:891      ESTABLISHED
[ir6.out] tcp        0      0 192.168.203.100:2049      192.168.203.153:690      ESTABLISHED
[ir11.out] tcp        0      0 192.168.204.100:2049      192.168.204.153:697      ESTABLISHED
[ir15.out] tcp        0      0 192.168.201.100:2049      192.168.201.153:943      ESTABLISHED
ir@SN1-FB-D1-27-2:/opt/ir/admin$
```

The following RMAN run block command was issued to restore the database from four mount points, /m01 to /m04.

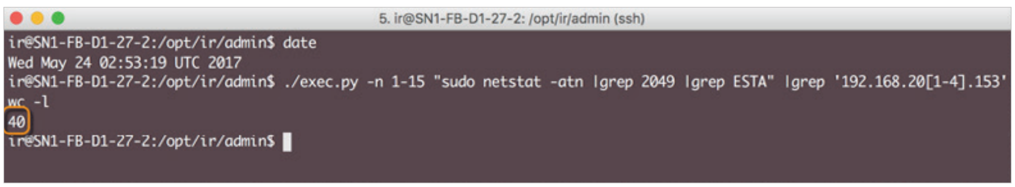
```
run
{
allocate channel c1 device type disk format '/m01/orclbkup/%U';
allocate channel c2 device type disk format '/m02/orclbkup/%U';
allocate channel c3 device type disk format '/m03/orclbkup/%U';
allocate channel c4 device type disk format '/m04/orclbkup/%U';
allocate channel c5 device type disk format '/m01/orclbkup/%U';
allocate channel c6 device type disk format '/m02/orclbkup/%U';
allocate channel c7 device type disk format '/m03/orclbkup/%U';
allocate channel c8 device type disk format '/m04/orclbkup/%U';
restore database;
recover database;
}
```

Once the restore command is started, checking the **v$dnfs_servers** shows that four filesystems (rman01 – rman04) were mounted.

```
SQL> select svrname, dirname, mntport, nfsport, wtmax, rtmax from v$dnfs _ servers;

SVRNAME        DIRNAME        MNTPORT    NFSPORT    WTMAX      RTMAX
-------------- ------------ ---------- ---------- ---------- ----------
FlashBlade     /rman01           2049       2049     524288     524288
FlashBlade     /rman03           2049       2049     524288     524288
FlashBlade     /rman04           2049       2049     524288     524288
FlashBlade     /rman02           2049       2049     524288     524288
```

Similarly to the backup activity, the restore also enabled multiple connections to FlashBlade from the host. Checking connection information on FlashBlade shows 40 connections spread across various blades.


```
5. ir@SN1-FB-D1-27-2: /opt/ir/admin (ssh)
ir@SN1-FB-D1-27-2:/opt/ir/admin$ date
Wed May 24 02:53:19 UTC 2017
ir@SN1-FB-D1-27-2:/opt/ir/admin$ ./exec.py -n 1-15 "sudo netstat -atn |grep 2049 |grep ESTA" |grep '192.168.20[1-4].153' |wc -l
40
ir@SN1-FB-D1-27-2:/opt/ir/admin$
```

Similarly, network traffic was spread across the four backup networks.

```
07:53:18 PM     IFACE   rxpck/s  txpck/s     rxkB/s    txkB/s rxcmp/s txcmp/s  rxmcst/s
07:53:19 PM   enp17s0 699712.00  49310.00 1039094.59  3608.32    0.00    0.00      0.00
07:53:19 PM   enp16s0 700475.00  46960.00 1040230.70  3446.96    0.00    0.00      0.00
07:53:19 PM    enp6s0      2.00      3.00       0.14     1.04    0.00    0.00      0.00
07:53:19 PM    enp7s0      0.00      0.00       0.00     0.00    0.00    0.00      0.00
07:53:19 PM        lo      0.00      0.00       0.00     0.00    0.00    0.00      0.00
07:53:19 PM    enp8s0 700461.00  48015.00 1040208.78  3518.31    0.00    0.00      0.00
07:53:19 PM   enp15s0 698943.00  47372.00 1037951.53  3474.84    0.00    0.00      0.00
```

The **v$session_longops** shows the details of the restores. Matching the previous recovery test with a single mount point, this test also took 323 seconds to restore 1.01 TB of data at a read rate of **3.2GB/s**.

```
SQL> select opname, start _ time, last _ update _ time, elapsed _ seconds, message
from v$session _ longops

OPNAME                       Start Time       Last Upd Time   ELAPSED _ SECONDS
---------------------------- ---------------- --------------- -------------------
MESSAGE
--------------------------------------------------------------------------------
RMAN: full datafile restore  05/23/17 19:50:27 05/23/17 19:54:44     257
RMAN: full datafile restore: Set Count 446: 19660800 out of 19660800 Blocks done

RMAN: full datafile restore  05/23/17 19:50:27 05/23/17 19:54:12     225
RMAN: full datafile restore: Set Count 449: 15728640 out of 15728640 Blocks done
```

```
RMAN: full datafile restore  05/23/17 19:51:22 05/23/17 19:55:38     256
RMAN: full datafile restore: Set Count 444: 19660800 out of 19660800 Blocks done

RMAN: full datafile restore  05/23/17 19:50:27 05/23/17 19:55:11     284
RMAN: full datafile restore: Set Count 447: 19660800 out of 19660800 Blocks done

RMAN: full datafile restore  05/23/17 19:50:27 05/23/17 19:55:06     279
RMAN: full datafile restore: Set Count 442: 19660800 out of 19660800 Blocks done

RMAN: full datafile restore  05/23/17 19:50:27 05/23/17 19:55:12     285
RMAN: full datafile restore: Set Count 445: 19660800 out of 19660800 Blocks done

RMAN: full datafile restore  05/23/17 19:50:27 05/23/17 19:55:10     283
RMAN: full datafile restore: Set Count 443: 19660800 out of 19660800 Blocks done

RMAN: aggregate input       05/23/17 19:50:26 05/23/17 19:55:49      323
RMAN: aggregate input: restore 460: 162171520 out of 162171520 Blocks done

RMAN: full datafile restore  05/23/17 19:50:27 05/23/17 19:54:28     241
RMAN: full datafile restore: Set Count 448: 16413440 out of 16413440 Blocks done

RMAN: full datafile restore  05/23/17 19:50:27 05/23/17 19:52:24     117
RMAN: full datafile restore: Set Count 450: 8132480 out of 8132480 Blocks done
```

The FlashBlade GUI shows a read bandwidth of around **4GB/s** for most of the restore activity, while the FlashArray GUI shows a write bandwidth in the range of **4GB/s** around the same time.

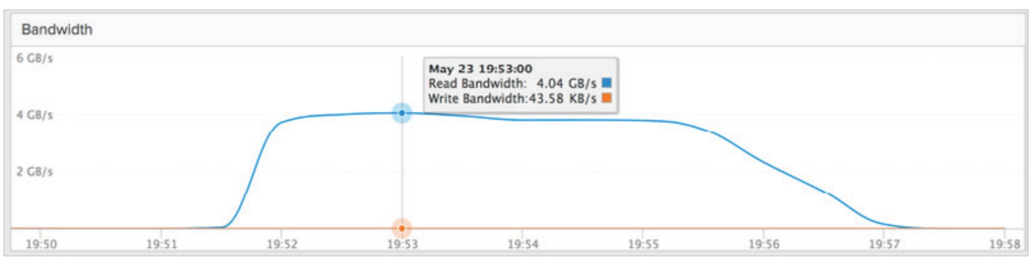


Figure 14. FlashBlade bandwidth

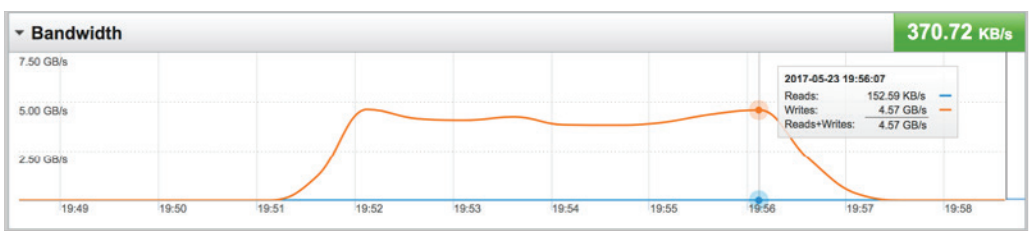


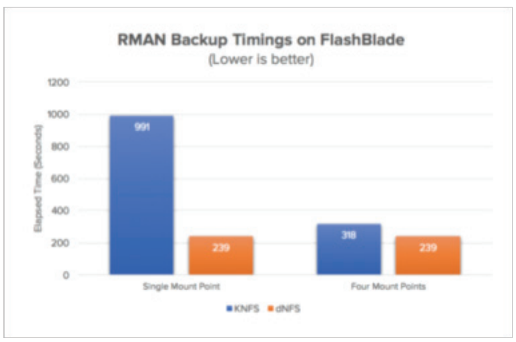
Figure 15. FlashArray bandwidth

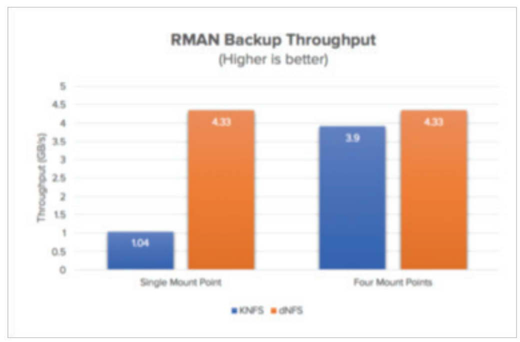
SUMMARY OF TEST RESULTS

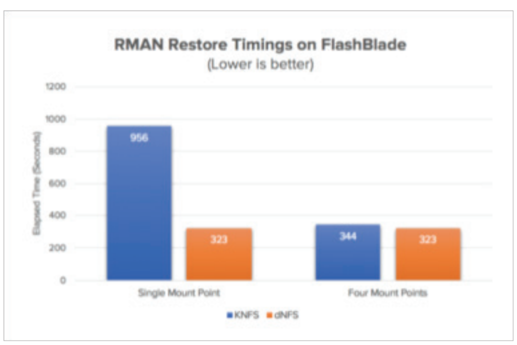
The various backup and recovery tests detailed above illustrate elapsed time and throughput to backup and restore a 1.01 TB database on FlashBlade using Kernel NFS and Direct NFS. Oracle implemented Direct NFS over Kernel NFS not just for manageability but also for performance, and that additional speed is complemented by the FlashBlade array's higher throughput.

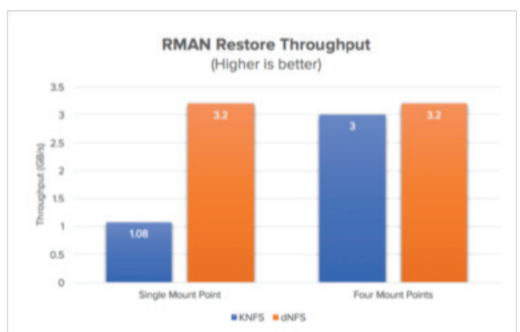
Use of a single mount point vs four mount points with Direct NFS on four subnets yields similar performance. Even in case of a single mount point, dNFS enables access to all the subnets and distributes traffic across all four interfaces, which quadruples performance as compared with the same test on Kernel NFS.

The following charts highlight the performance of FlashBlade across each of the backup and recovery tests.









From these results, it's clear that dNFS with FlashBlade gives higher throughput, with the additional benefits of high availability and multipathing-like behavior. Even Kernel NFS with multiple mount points yielded better throughput numbers in comparison to any other solution on the market.

BEST PRACTICES FOR ORACLE ON FLASHBLADE

NFS FILESYSTEMS FOR ORACLE

Networked filesystems provide the flexibility to mount the same filesystem across multiple hosts to enable shared storage over TCP/IP. Hence, NFS filesystem is a viable option for an Oracle RAC environment that requires shared storage across all RAC nodes. Also, for customers who have a standardized Ethernet-based connection between database hosts and storage instead of Fiber Channel, NAS storage systems like FlashBlade that provide networked filesystems would be the preferred choice. Make sure the NFS protocol is selected when creating the filesystem on FlashBlade.

USE dNFS OVER KERNEL NFS

To scale up bandwidth on FlashBlade, enable numerous connections from the client rather than a single connection. Oracle's Direct NFS creates a separate connection to the storage system for every server process, as opposed to a single connection per mount point with Kernel NFS.

ENABLE PARALLELISM

To increase read and write bandwidth on FlashBlade, use client level parallelization techniques like parallel queries and multiple RMAN channels based on the CPU availability of your host in conjunction with dNFS. This increases the number of connections to FlashBlade, especially with dNFS.

USE MULTIPLE NETWORK INTERFACES

To enhance network bandwidth, be sure to have multiple network interfaces on the client. These multiple interfaces can be configured on a single subnet or on multiple subnets.

SINGLE SUBNET

IO performance in a dNFS environment with multiple interfaces and a single subnet is limited to the speed of the first interface that is picked by the OS. This is because the OS returns the first path when multiple paths are available on the subnet, hence traffic is always routed through the first path.

For example, in the setup below, FlashBlade has a single IP address on which the NFS filesystem will be mounted from the client, and the client has two interfaces.

```
Client interface        NFS server
ens7f0  10.21.108.193    10.21.108.10
ens7f1  10.21.108.194    10.21.108.10
```

```
[oracle@rlxora-b01-11-06 ~]$ route -n
Kernel IP routing table
```

```
Destination   Gateway     Genmask         Flags Metric Ref Use Iface
0.0.0.0       10.21.108.1  0.0.0.0         UG     0       0  0   ens7f0
10.21.108.0   0.0.0.0      255.255.255.0  U      0       0  0   ens7f0 <-- first route
10.21.108.0   0.0.0.0      255.255.255.0  U      0       0  0   ens7f1 <-- route ignored
```

As per the routing table, both traffic can go through the first interface (ens7f0) as the destination; the mask fits for both routes and the OS will invariably choose the first route.

In this case, to enhance bandwidth, NIC bonding at the client level is recommended. An Oracle support document (Doc ID 833481.1) provides other means to enable multiple paths in a single subnet using static routing, but it doesn't address the issue of availability when a NIC fails or a network cable is pulled, as the routing table will not be updated.

MULTIPLE SUBNETS

"Direct NFS client best practices recommend always to use multipaths in separate subnets."

Oracle recommends using a separate subnet for each interface and supports up to four subnets. With multiple subnets, there is no need to bond the network interfaces to aggregate bandwidth across the available interfaces. Routing will be automatic in the case of multiple subnets.

```
Client interface       NFS server
ens7f0 10.21.108.193    10.21.108.10
ens7f1 10.21.107.194    10.21.107.10
```

```
[oracle@rlxora-b01-11-06 ~]$ route -n
Kernel IP routing table
Destination     Gateway        Genmask        Flags Metric Ref    Use Iface
0.0.0.0         10.21.108.1     0.0.0.0         UG     0       0       0   ens7f0
10.21.108.0     *               255.255.255.0   U      0       0       0   ens7f0
10.21.107.0     *               255.255.255.0   U      0       0       0   ens7f1
```

In this case, these are two dynamic routes and, based on traffic, the route is selected automatically.

As such, if you decide to use multiple subnets, they should be configured on both the client and the FlashBlade side. Multiple subnets can be configured in the FlashBlade GUI under Network Settings.

Be sure to update the **oranfstab** with the subnets and the mount point details. In an RAC environment, all RAC nodes should have the appropriate **oranfstab** file configured.

NFS VOLUMES AND MOUNTPOINT REQUIREMENTS

It is not required to have as many NFS filesystems/volumes as subnets for dNFS to be effective. Nor is it required to mount a single volume on to all subnets for dNFS to be effective. Oracle dNFS reads the **oranfstab** and based on the storage paths and mount details, it will create multiple paths when the database files are accessed.

For example, with two subnets and two mounts, dNFS would create four paths to the storage system for every server process.

LINUX MOUNT OPTIONS

To mount the NFS filesystem on Linux, use the following mount options. Do not specify the rsize or wsize options, as the system can get the default offered by FlashBlade, which is 524288).

rw, bg, nointr, hard, tcp, vers=3, nolock, noac, actimeo=0

ENABLING & DISABLING DIRECT NFS CLIENT

To enable the Direct NFS Client, the standard Oracle Disk Manager (ODM) library that supports the Direct NFS Client should be used. It can be enabled as follows (starting Oracle 11.2).

```
cd $ORACLE_HOME/rdbms/lib
make -f ins_rdbms.mk dnfs_on
```

To disable the Direct NFS Client, perform the following.

```
cd $ORACLE_HOME/rdbms/lib
make -f ins_rdbms.mk dnfs_off
```

VERIFYING THE USE OF DIRECT NFS CLIENT

1) If dNFS is enabled, the alert.log will show the following entry when the database is started.

```
Oracle instance running with ODM: Oracle Direct NFS ODM Library Version 4.0
```

2) Check the dNFS server information from v$dnfs_servers view.

```
SQL> select svrname, dirname, mntport, nfsport, wtmax, rtmax from v$dnfs_servers;
SVRNAME            DIRNAME                 MNTPORT    NFSPORT      WTMAX      RTMAX
------------------ -------------------- ---------- ---------- ---------- ----------
FlashBlade         /rman01                    2049       2049     524288     524288
```

Note: Even though dNFS is enabled, Oracle only mounts the volume/filesystem and opens the files when they are accessed. If no datafiles are accessed, then the above view will return no rows.

CONCLUSION

Businesses need a scalable and performant storage system to meet their recovery time objectives. Pure Storage FlashBlade provides a scalable, highly available, data reducible, and performant solution to meet Oracle RMAN backup requirements. Our various backup tests using Oracle RMAN with dNFS enabled illustrate the highly performant write rates of FlashBlade, with peaks over 4GB/s. Having FlashBlade as the backup target alone cannot provide these outstanding results – the source system should be capable of reading and writing data out and in at required rates, without which, the source system becomes the bottleneck. Complementing FlashBlade, the Pure FlashArray system provides scalable read bandwidth for backup activity and comparable write bandwidth for restore activity.

Oracle dNFS provides highly-tuned NFS performance over kernel NFS. It's free and provides fault tolerance and high availability with simplified management. Using dNFS under the architecture of FlashBlade provides higher sustained read and write bandwidths. These bandwidths would be more than sufficient for even larger organizations with varying amounts of incremental changes in their database environment and helps significantly in meeting enterprise time objectives either with recovery or with service levels.

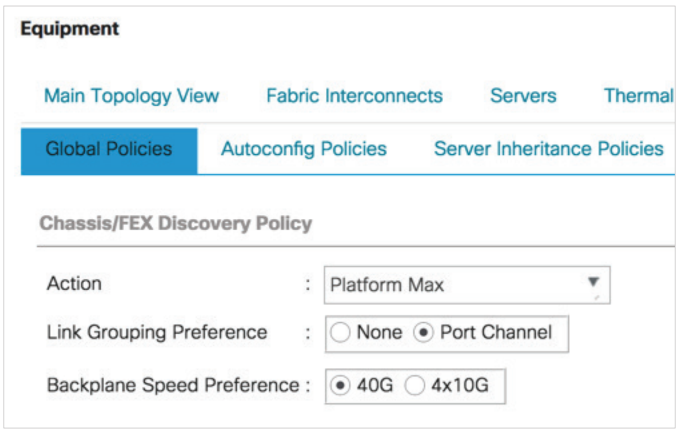
REFERENCES

Following documents and links were referred to in preparing this document.

1. Pure Storage Community pages
   https://support.purestorage.com
2. Oracle Support KB article "Direct NFS: FAQ Doc ID 954425.1"
   https://support.oracle.com

APPENDIX A: NETWORK SPEED PREFERENCE ON CISCO UCS

Cisco UCS offers the option of selecting a backplane speed of either 40G or 4x10G at the global policy level. Please exercise caution – changing this policy can be intrusive, as it requires re-acknowledgement of all servers and possible reboot of all of them.



The **ethtool** command in Linux shows the speed at 40Gb/s for every interface that is configured out of VIC 1340. Sample output from one of the interfaces is given below. Using this option, a single subnet with dNFS can achieve a bandwidth up to 40Gb/s.

```
[oracle@olxora01 ~]$ ethtool enp8s0
Settings for enp8s0:
        Supported ports: [ FIBRE ]
        Supported link modes:   10000baseT/Full
        Supported pause frame use: No
        Supports auto-negotiation: No
        Advertised link modes:  10000baseT/Full
        Advertised pause frame use: No
        Advertised auto-negotiation: No
        Speed: 40000Mb/s
        Duplex: Full
        Port: FIBRE
        PHYAD: 0
        Transceiver: external
        Auto-negotiation: off
```


ABOUT THE AUTHOR

Somu Rajarathinam is the Pure Storage Oracle Solutions Architect responsible for defining database solutions based on the company's products, performing benchmarks, and developing reference architectures for Oracle databases on Pure.

Somu has over 20 years of Oracle database experience, including as a member of Oracle Corporation's Systems Performance and Oracle Applications Performance Groups. His career has also included assignments with Logitech, Inspirage, and Autodesk, ranging from providing database and performance solutions, to managing Infrastructure, to delivering database and application support, both in-house and in the cloud.

Web: **www.somu.us**

Twitter: **@purelydb**

ps_ra_oracle-rman-backup-and-recovery-on-flashblade_01

Pure Storage, Inc.
Twitter: @purestorage
www.purestorage.com

650 Castro Street, Suite #260
Mountain View, CA 94041

T: 650-290-6088
F: 650-625-9667

Sales: sales@purestorage.com