Host configuration with ASA r2 systems

Enterprise applications

NetApp
February 10, 2026

Table of Contents

Host configuration with ASA r2 systems

AIX

Configuration topics for Oracle database on IBM AIX with ASA r2 ONTAP.

AIX is supported with NetApp ASA r2 for hosting Oracle databases, provided:

- You configure Oracle for concurrent I/O properly.
- You use supported SAN protocols (FC/iSCSI/NVMe).
- You run ONTAP 9.16.x or later on ASA r2.

Concurrent I/O

Achieving optimum performance on IBM AIX with ASA r2 requires the use of concurrent I/O. Without concurrent I/O, performance limitations are likely because AIX performs serialized, atomic I/O, which incurs significant overhead.

Originally, NetApp recommended using the `cio` mount option to force concurrent I/O on the file system, but this process had drawbacks and is no longer required. Since the introduction of AIX 5.2 and Oracle 10gR1, Oracle on AIX can open individual files for concurrent I/O, as opposed to forcing concurrent I/O on the entire file system.

The best method for enabling concurrent I/O is to set the `init.ora` parameter `filesystemio_options` to `setall`. Doing so allows Oracle to open specific files for use with concurrent I/O.

Using cio as a mount option forces the use of concurrent I/O, which can have negative consequences. For example, forcing concurrent I/O disables readahead on file systems, which can damage performance for I/O occurring outside the Oracle database software, such as copying files and performing tape backups. Furthermore, products such as Oracle GoldenGate and SAP BR*Tools are not compatible with using the cio mount option with certain versions of Oracle.

NetApp recommends the following:

- Do not use the `cio` mount option at the file system level. Rather, enable concurrent I/O through the use of `filesystemio_options=setall`.
- Only use the `cio` mount option if it is not possible to set `filesystemio_options=setall`.

Since ASA r2 does not support NAS, all Oracle deployments on AIX must use block protocols.

AIX jfs/jfs2 Mount Options

The following table lists the AIX jfs/jfs2 mount options.

File type	Mount options
ADR Home	Defaults

File type	Mount options
Controlfiles	Defaults
Datafiles	Defaults
Redo logs	Defaults
ORACLE_HOME	Defaults

Before using AIX `hdisk` devices in any environment, including databases, check the parameter `queue_depth`. This parameter is not the HBA queue depth; rather it relates to the SCSI queue depth of the individual `hdisk device`. Depending on how the ASA r2 LUNs are configured, the value for `queue_depth` might be too low for good performance. Testing has shown the optimum value to be 64.

HP-UX

Configuration topics for Oracle database on HP-UX with ASA r2 ONTAP.

HP-UX is supported with NetApp ASA r2 for hosting Oracle databases, provided:

- ONTAP version is 9.16.x or later.
- Use SAN protocols (FC/iSCSI/NVMe). NAS is not supported on ASA r2.
- Apply HP-UX-specific mount and I/O tuning best practices.

HP-UX VxFS mount options

Use the following mount options for file systems hosting Oracle binaries:

```
delaylog,nodatainlog
```

Use the following mount options for file systems containing datafiles, redo logs, archive logs, and control files in which the version of HP-UX does not support concurrent I/O:

```
nodatainlog,mincache=direct,convosync=direct
```

When concurrent I/O is supported (VxFS 5.0.1 and later, or with the ServiceGuard Storage Management Suite), use these mount options for file systems containing datafiles, redo logs, archive logs, and control files:

```
delaylog,cio
```

The parameter `db_file_multiblock_read_count` is especially critical in VxFS environments. Oracle recommends that this parameter remain unset in Oracle 10g R1 and later unless specifically directed otherwise. The default with an Oracle 8KB block size is 128. If the value of this parameter is forced to 16 or less, remove the `convosync=direct` mount option because it can damage sequential I/O performance. This step damages other aspects of performance and should only be taken if the value of `db_file_multiblock_read_count` must be changed from the default value.

Linux

Configuration topics specific to the Linux OS with ASA r2 ONTAP.

Linux (Oracle Linux, RHEL, SUSE) is supported with ASA r2 for Oracle databases. Use SAN protocols, configure multipathing correctly, and apply Oracle best practices for ASM and I/O tuning.

I/O scheduler

The Linux kernel allows low-level control over the way that I/O to block devices is scheduled. The defaults on various distribution of Linux vary considerably. Testing shows that Deadline usually offers the best results, but on occasion NOOP has been slightly better. The difference in performance is minimal, but test both options if it is necessary to extract the maximum possible performance from a database configuration. CFQ is the default in many configurations, and it has demonstrated significant performance problems with database workloads.

See the relevant Linux vendor documentation for instructions on configuring the I/O scheduler.

Multipathing

Some customers have encountered crashes during network disruption because the multipath daemon was not running on their system. On recent versions of Linux, the installation process of the OS and the multipathing daemon might leave these OSs vulnerable to this problem. The packages are installed correctly, but they are not configured for automatic startup after a reboot.

For example, the default for the multipath daemon on RHEL 9.7 might appear as follows:

```
[root@host1 ~]# systemctl list-unit-files --type=service | grep multipathd
multipathd.service                          disabled
```

This can be corrected with the following commands:

```
[root@host1 ~]# systemctl enable multipathd.service
[root@host1 ~]# systemctl list-unit-files --type=service | grep multipathd
multipathd.service                          enabled
```

Queue depth

Set appropriate queue depth for SAN devices to avoid I/O bottlenecks. The default queue depth on Linux is often set to 128, which can lead to performance problems with Oracle databases. Setting the queue depth too high can cause excessive I/O queuing, leading to increased latency and reduced throughput. Setting it too low can limit the number of outstanding I/O requests, reducing overall performance. A queue depth of 64 is often a good starting point for Oracle database workloads on ASA r2, but it may need to be adjusted based on specific workload characteristics and performance testing.

ASM mirroring

ASM mirroring might require changes to the Linux multipath settings to allow ASM to recognize a problem and switch over to an alternate fail group. Most ASM configurations on ONTAP use external redundancy, which means that data protection is provided by the external array and ASM does not mirror data. Some sites use ASM with normal redundancy to provide two-way mirroring, normally across different sites.

For ASA r2 systems that support active-active multipathing, these multipath settings should be adjusted. Since all paths are active and load-balanced, indefinite queuing is not required. Instead, multipath parameters should prioritize performance and quick failback. This behavior is important for ASM mirroring because ASM must receive an I/O failure for it to retry I/O on an alternate LUN. If I/O is queued indefinitely, ASM cannot trigger a failover.

Set the following parameters in the Linux `multipath.conf` file for ASM LUNs used with ASM mirroring:

```
polling_interval 5
no_path_retry 24
failback immediate
path_grouping_policy multibus
path_selector "service-time 0"
```

These settings create a 120-second timeout for ASM devices. The timeout is calculated as the `polling_interval` * `no_path_retry` as seconds. The exact value might need to be adjusted in some circumstances, but a 120 second timeout should be sufficient for most uses. Specifically, 120 seconds should allow a controller takeover or giveback to occur without producing an I/O error that would result in the fail group being taken offline.

A lower `no_path_retry` value can shorten the time required for ASM to switch to an alternate fail group, but this also increases the risk of an unwanted failover during maintenance activities such as a controller takeover. The risk can be mitigated by careful monitoring of the ASM mirroring state. If an unwanted failover occurs, the mirrors can be rapidly resynced if the resync is performed relatively quickly. For additional information, see the Oracle documentation on ASM Fast Mirror Resync for the version of Oracle software in use.

Linux xfs, ext3, and ext4 mount options

NetApp recommends using the default mount options. Ensure proper alignment when creating file systems on LUNs.

ASMLib/AFD (ASM Filter Driver)

Configuration topics specific to the Linux OS using AFD and ASMlib with ASA r2 ONTAP.

ASMlib block sizes

ASMlib is an optional ASM management library and associated utilities. Its primary value is the capability to stamp a LUN as an ASM resource with a human-readable label.

Recent versions of ASMlib detect a LUN parameter called Logical Blocks Per Physical Block Exponent (LBPPBE). This value was not reported by the ONTAP SCSI target until recently. It now returns a value that indicates that a 4KB block size is preferred. This is not a definition of block size, but it is a hint to any application that uses LBPPBE that I/Os of a certain size might be handled more efficiently. ASMlib does, however, interpret LBPPBE as a block size and persistently stamps the ASM header when the ASM device is created.

This process can cause problems with upgrades and migrations in a number of ways, all based on the inability to mix ASMlib devices with different block sizes in the same ASM diskgroup.

For example, older arrays generally reported an LBPPBE value of 0 or did not report this value at all. ASMlib interprets this as a 512-byte block size. Newer arrays would be interpreted as having a 4KB block size. It is not possible to mix both 512-byte and 4KB devices in the same ASM diskgroup. Doing so would block a user from increasing the size of the ASM diskgroup using LUNs from two arrays or leveraging ASM as a migration tool. In other cases, RMAN might not permit the copying of files between an ASM diskgroup with a 512-byte block size and an ASM diskgroup with a 4KB block size.

The preferred solution is to patch ASMlib. The Oracle bug ID is 13999609, and the patch is present in oracleasm-support-2.1.8-1 and higher. This patch allows a user to set the parameter `ORACLEASM_USE_LOGICAL_BLOCK_SIZE` to `true` in the `/etc/sysconfig/oracleasm` configuration file. Doing so blocks ASMlib from using the LBPPBE parameter, which means that LUNs on the new array are now recognized as 512-byte block devices.

> The option does not change the block size on LUNs that were previously stamped by ASMlib. For example, if an ASM diskgroup with 512-byte blocks must be migrated to a new storage system that reports a 4KB block, the option `ORACLEASM_USE_LOGICAL_BLOCK_SIZE` must be set before the new LUNs are stamped with ASMlib. If devices have already been stamped by oracleasm, they must be reformatted before being restamped with a new block size. First, deconfigure the device with `oracleasm deletedisk`, and then clear the first 1GB of the device with `dd if=/dev/zero of=/dev/mapper/device bs=1048576 count=1024`. Finally, if the device had been previously partitioned, use the `kpartx` command to remove stale partitions or simply reboot the OS.

If ASMlib cannot be patched, ASMlib can be removed from the configuration. This change is disruptive and requires the unstamping of ASM disks and making sure that the `asm_diskstring` parameter is set correctly. This change does not, however, require the migration of data.

ASM Filter Drive (AFD) block sizes

AFD is an optional ASM management library which is becoming the replacement for ASMlib. From a storage point of view, it is very similar to ASMlib, but it includes additional features such as the ability to block non-Oracle I/O to reduce the chances of user or application errors that could corrupt data.

Device block sizes

Like ASMlib, AFD also reads the LUN parameter Logical Blocks Per Physical Block Exponent (LBPPBE) and by default uses the physical block size, not the logical block size.

This could create a problem if AFD is added to an existing configuration where the ASM devices are already formatted as 512 byte block devices. The AFD driver would recognize the LUN as a 4K device and the mismatch between the ASM label and the physical device would prevent access. Likewise, migrations would be affected because it is not possible to mix both 512-byte and 4KB devices in the same ASM diskgroup. Doing so would block a user from increasing the size of the ASM diskgroup using LUNs from two arrays or leveraging ASM as a migration tool. In other cases, RMAN might not permit the copying of files between an ASM diskgroup with a 512-byte block size and an ASM diskgroup with a 4KB block size.

The solution is simple - AFD includes a parameter to control whether it uses the logical or physical block sizes. This is a global parameter affecting all devices on the system. To force AFD to use the logical block size, set `options oracleafd oracleafd_use_logical_block_size=1` in the `/etc/modprobe.d/oracleafd.conf` file.

Multipath transfer sizes

Recent linux kernel changes enforce I/O size restrictions sent to multipath devices, and AFD does not honor these restrictions. The I/Os are then rejected, which causes the LUN path to go offline. The result is an inability to install Oracle Grid, configure ASM, or create a database.

The solution is to manually specify the maximum transfer length in the multipath.conf file for ONTAP LUNs:

```
devices {
            device {
                vendor "NETAPP"
                product "LUN.*"
                max_sectors_kb 4096
        }
    }
```

> Even if no problems currently exist, this parameter should be set if AFD is used to ensure that a future linux upgrade does not unexpectedly cause problems.

Microsoft Windows

Configuration topics for Oracle database on Microsoft Windows with ASA r2 ONTAP.

SAN

For optimal compression efficiency, ensure the NTFS file system uses an 8K or larger allocation unit. Use of a 4K allocation unit, which is generally the default, negatively impacts compression efficiency.

Solaris

Configuration topics specific to the Solaris OS with ASA r2 ONTAP.

Solaris UFS mount options

NetApp strongly recommends using the logging mount option so that data integrity is preserved in the case of a Solaris host crash or the interruption of FC connectivity. The logging mount option also preserves the usability of Snapshot backups.

Solaris ZFS

Solaris ZFS must be installed and configured carefully to deliver optimum performance.

mvector

Solaris 11 included a change in how it processes large I/O operations which can result in severe performance problems on SAN storage arrays. The problem is documented NetApp tracking bug report 630173, "Solaris 11 ZFS Performance Regression."

This is not an ONTAP bug. It is a Solaris defect that is tracked under Solaris defects 7199305 and 7082975.

You can consult Oracle Support to find out if your version of Solaris 11 is affected, or you can test the workaround by changing `zfs_mvector_max_size` to a smaller value.

You can do this by running the following command as root:

```
[root@host1 ~]# echo "zfs_mvector_max_size/W 0t131072" |mdb -kw
```

If any unexpected problems arise from this change, it can be easily reversed by running the following command as root:

```
[root@host1 ~]# echo "zfs_mvector_max_size/W 0t1048576" |mdb -kw
```

Kernel

Reliable ZFS performance requires a Solaris kernel patched against LUN alignment problems. The fix was introduced with patch 147440-19 in Solaris 10 and with SRU 10.5 for Solaris 11. Only use Solaris 10 and later with ZFS.

LUN configuration

To configure a LUN, complete the following steps:

1. Create a LUN of type `solaris`.
2. Install the appropriate Host Utility Kit (HUK) specified by the NetApp Interoperability Matrix Tool (IMT).
3. Follow the instructions in the HUK exactly as described. The basic steps are outlined below, but refer to the latest documentation for the proper procedure.
   a. Run the `host_config` utility to update the `sd.conf/sdd.conf` file. Doing so allows the SCSI drives to correctly discover ONTAP LUNs.
   b. Follow the instructions given by the `host_config` utility to enable multipath input/output (MPIO).
   c. Reboot. This step is required so that any changes are recognized across the system.
4. Partition the LUNs and verify that they are properly aligned. See "Appendix B: WAFL Alignment Verification” for instructions on how to directly test and confirm alignment.

zpools

A zpool should only be created after the steps in the LUN Configuration are performed. If the procedure is not done correctly, it can result in serious performance degradation due to the I/O alignment. Optimum performance on ONTAP requires I/O to be aligned to a 4K boundary on a drive. The file systems created on a zpool use an effective block size that is controlled through a parameter called `ashift`, which can be viewed by running the command `zdb -C`.

The value of `ashift` defaults to 9, which means 2^9, or 512 bytes. For optimum performance, the `ashift` value must be 12 (2^12=4K). This value is set at the time the zpool is created and cannot be changed, which means that data in zpools with `ashift` other than 12 should be migrated by copying data to a newly created zpool.

After creating a zpool, verify the value of `ashift` before proceeding. If the value is not 12, the LUNs were not discovered correctly. Destroy the zpool, verify that all steps shown in the relevant Host Utilities documentation were performed correctly, and recreate the zpool.

zpools and Solaris LDOMs

Solaris LDOMs create an additional requirement for making sure that I/O alignment is correct. Although a LUN might be properly discovered as a 4K device, a virtual vdsk device on an LDOM does not inherit the configuration from the I/O domain. The vdsk based on that LUN defaults back to a 512-byte block.

An additional configuration file is required. First, the individual LDOM's must be patched for Oracle bug 15824910 to enable the additional configuration options. This patch has been ported into all currently used versions of Solaris. Once the LDOM is patched, it is ready for configuration of the new properly aligned LUNs as follows:

1. Identify the LUN or LUNs to be used in the new zpool. In this example, it is the c2d1 device.

```
[root@LDOM1 ~]# echo | format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c2d0 <Unknown-Unknown-0001-100.00GB>
     /virtual-devices@100/channel-devices@200/disk@0
  1. c2d1 <SUN-ZFS Storage 7330-1.0 cyl 1623 alt 2 hd 254 sec 254>
     /virtual-devices@100/channel-devices@200/disk@1
```

2. Retrieve the vdc instance of the devices to be used for a ZFS pool:

```
[root@LDOM1 ~]#  cat /etc/path_to_inst
#
# Caution! This file contains critical kernel state
#
"/fcoe" 0 "fcoe"
"/iscsi" 0 "iscsi"
"/pseudo" 0 "pseudo"
"/scsi_vhci" 0 "scsi_vhci"
"/options" 0 "options"
"/virtual-devices@100" 0 "vnex"
"/virtual-devices@100/channel-devices@200" 0 "cnex"
"/virtual-devices@100/channel-devices@200/disk@0" 0 "vdc"
"/virtual-devices@100/channel-devices@200/pciv-communication@0" 0 "vpci"
"/virtual-devices@100/channel-devices@200/network@0" 0 "vnet"
"/virtual-devices@100/channel-devices@200/network@1" 1 "vnet"
"/virtual-devices@100/channel-devices@200/network@2" 2 "vnet"
"/virtual-devices@100/channel-devices@200/network@3" 3 "vnet"
"/virtual-devices@100/channel-devices@200/disk@1" 1 "vdc" << We want
this one
```

3. Edit `/platform/sun4v/kernel/drv/vdc.conf`:

```
block-size-list="1:4096";
```

This means that device instance 1 is assigned a block size of 4096.

As an additional example, assume vdsk instances 1 through 6 need to be configured for a 4K block size and `/etc/path_to_inst` reads as follows:

```
"/virtual-devices@100/channel-devices@200/disk@1" 1 "vdc"
"/virtual-devices@100/channel-devices@200/disk@2" 2 "vdc"
"/virtual-devices@100/channel-devices@200/disk@3" 3 "vdc"
"/virtual-devices@100/channel-devices@200/disk@4" 4 "vdc"
"/virtual-devices@100/channel-devices@200/disk@5" 5 "vdc"
"/virtual-devices@100/channel-devices@200/disk@6" 6 "vdc"
```

4. The final `vdc.conf` file should contain the following:

```
block-size-list="1:8192","2:8192","3:8192","4:8192","5:8192","6:8192";
```

> ! The LDOM must be rebooted after vdc.conf is configured and the vdsk is created. This step cannot be avoided. The block size change only takes effect after a reboot. Proceed with zpool configuration and ensure that ashift is properly set to 12 as described previously.

ZFS Intent Log (ZIL)

Generally, there is no reason to locate the ZFS Intent Log (ZIL) on a different device. The log can share space with the main pool. The primary use of a separate ZIL is when using physical drives that lack the write caching features in modern storage arrays.

logbias

Set the `logbias` parameter on ZFS file systems hosting Oracle data.

```
zfs set logbias=throughput <filesystem>
```

Using this parameter reduces overall write levels. Under the defaults, written data is committed first to the ZIL and then to the main storage pool. This approach is appropriate for a configuration using a plain drive configuration, which includes an SSD-based ZIL device and spinning media for the main storage pool. This is because it allows a commit to occur in a single I/O transaction on the lowest latency media available.

When using a modern storage array that includes its own caching capability, this approach is not generally necessary. Under rare circumstances, it might be desirable to commit a write with a single transaction to the log, such as a workload that consists of highly concentrated, latency-sensitive random writes. There are consequences in the form of write amplification because the logged data is eventually written to the main storage pool, resulting in a doubling of the write activity.

Direct I/O

Many applications, including Oracle products, can bypass the host buffer cache by enabling direct I/O. This strategy does not work as expected with ZFS file systems. Although the host buffer cache is bypassed, ZFS itself continues to cache data. This action can result in misleading results when using tools such as fio or sio to perform performance tests because it is difficult to predict whether I/O is reaching the storage system or whether it is being cached locally within the OS. This action also makes it very difficult to use such synthetic tests to compare ZFS performance to other file systems. As a practical matter, there is little to no difference in file system performance under real user workloads.

Multiple zpools

Snapshot-based backups, restores, clones, and archiving of ZFS-based data must be performed at the level of the zpool and typically requires multiple zpools. A zpool is analogous to an LVM disk group and should be configured using the same rules. For example, a database is probably best laid out with the datafiles residing on `zpool1` and the archive logs, control files, and redo logs residing on `zpool2`. This approach permits a standard hot backup in which the database is placed in hot backup mode, followed by a snapshot of `zpool1`. The database is then removed from hot backup mode, the log archive is forced, and a snapshot of `zpool2` is created. A restore operation requires unmounting the zfs file systems and offlining the zpool in its entirety, following by a SnapRestore restore operation. The zpool can then be brought online again and the database recovered.

filesystemio_options

The Oracle parameter `filesystemio_options` works differently with ZFS. If `setall` or `directio` is used, write operations are synchronous and bypass the OS buffer cache, but reads are buffered by ZFS. This action causes difficulties in performance analysis because I/O is sometimes intercepted and serviced by the ZFS cache, making storage latency and total I/O less than it might appear to be.