BEST PRACTICES GUIDE

# Nimble Storage for Oracle Database on OL6 & RHEL6 with Fibre Channel or iSCSI

## Document Revision

**Table 1.**

| Date | Revision | Description |
|---|---|---|
| 1/9/2012 | 1.0 | Initial Draft |
| 7/2/2013 | 1.1 | Revised |
| 3/12/2014 | 1.2 | Revised iSCSI Setting |
| 5/8/2014 | 1.3 | Added ASM AU Size |
| 11/10/2014 | 1.4 | Updated iSCSI and Multipath |
| 1/30/2015 | 1.5 | Added Fibre Channel |

# Table of Contents

## *Introduction*

The purpose of this technical white paper is to walk through the step-by-step for tuning Oracle databases on Nimble Storage running on Oracle Linux or Red Hat Linux operating system.

## *Audience*

This guide is intended for Oracle database solution architects, storage engineers, system administrators and IT managers who analyze, design and maintain a robust database environment on Nimble Storage. It is assumed that the reader has a working knowledge of iSCSI/FC SAN network design, and basic Nimble Storage operations. Knowledge of Oracle Linux operating system, Oracle Clusterware, and Oracle database is also required.

## *Scope*

During the design phase for a new Oracle database implementation, DBAs and Storage Administrators often times work together to come up with the best storage needs. They have to consider many storage configuration options to facilitate high performance and high availability. In order to protect data against failures of disk drives, host bus adapters (HBAs), and switches, they need to consider using different RAID levels and multiple paths. When you have different RAID levels come into play for performance, TCO tends to increase as well. For example, in order to sustain a certain number of IOPS with low latency for an OLTP workload, DBAs would require a certain number of 15K disk drives with RAID 10. The higher the number of required IOPS, the more 15K drives are needed. The reason is because mechanical disk drives have seek times and transfer rate, therefore, you would need more of them to handle the required IOPS with acceptable latency. This will increase the TCO tremendously over time. Not to mention that if the database is small in capacity but the required IOPS is high, you would end up with a lot of wasted space in your SAN.

This white paper explains the Nimble technology and how it can lower the TCO of your Oracle environment and still achieve the performance required. This paper also discusses the best practices for implementing Oracle databases on Nimble Storage.

# Nimble Storage Features

## Cache Accelerated Sequential Layout (CASL™)

Nimble Storage solutions are built on its patented Cache Accelerated Sequential Layout (CASL™) architecture. CASL leverages the unique properties of flash and disk to deliver high performance and capacity – all within a dramatically small footprint.

CASL and InfoSight™ form the foundation of the Adaptive Flash platform, which allows for the dynamic and intelligent deployment of storage resources to meet the growing demands of business-critical applications.

### Dynamic Flash-Based Read Caching

CASL caches "hot" active data onto SSD in real time—without the need to set complex policies. This way it can instantly respond to read requests—as much as 10X faster than traditional bolt-on or tiered approach to flash.

### Write-Optimized Data Layout

CASL collects or coalesces random writes, compresses them, and writes them sequentially to disks. This results in write operations that are as much as 100x faster than traditional disk-based storage.

### Inline Compression

CASL compresses data as it is written to the array with no performance impact. It takes advantage of efficient variable block compression and multicore processors. A recent measurement of our installed base shows average compression rates from 30 to 75 percent for a variety of workloads.

### Scale-to-Fit Flexibility

CASL allows for the non-disruptive and independent scaling of performance and capacity. This is accomplished by either upgrading the storage controller (compute) for higher throughput, moving to larger flash SSD (cache) to accommodate more active data, or by adding storage shelves to boost capacity. This flexible scaling eliminates the need for disruptive forklift upgrades.

### Snapshots and Integrated Data Protection

CASL can take thousands of point-in-time instant snapshots of volumes by creating a copy of the volumes' indices. Any updates to existing data or new data written to a volume are redirected to free space (optimized by CASL's unique data layout). This means there is no performance impact due to snapshots and snapshots take little incremental space as only changes are maintained. This also simplifies restoring snapshots, as no data needs to be copied.

### Efficient Integrated Replication

Nimble Storage efficiently replicates data to another array by transferring compressed, block-level changes only. These remote copies can be made active if the primary array becomes unavailable. This makes deploying

disaster data recovery easy and affordable – especially over a WAN to a remote array where bandwidth is limited.

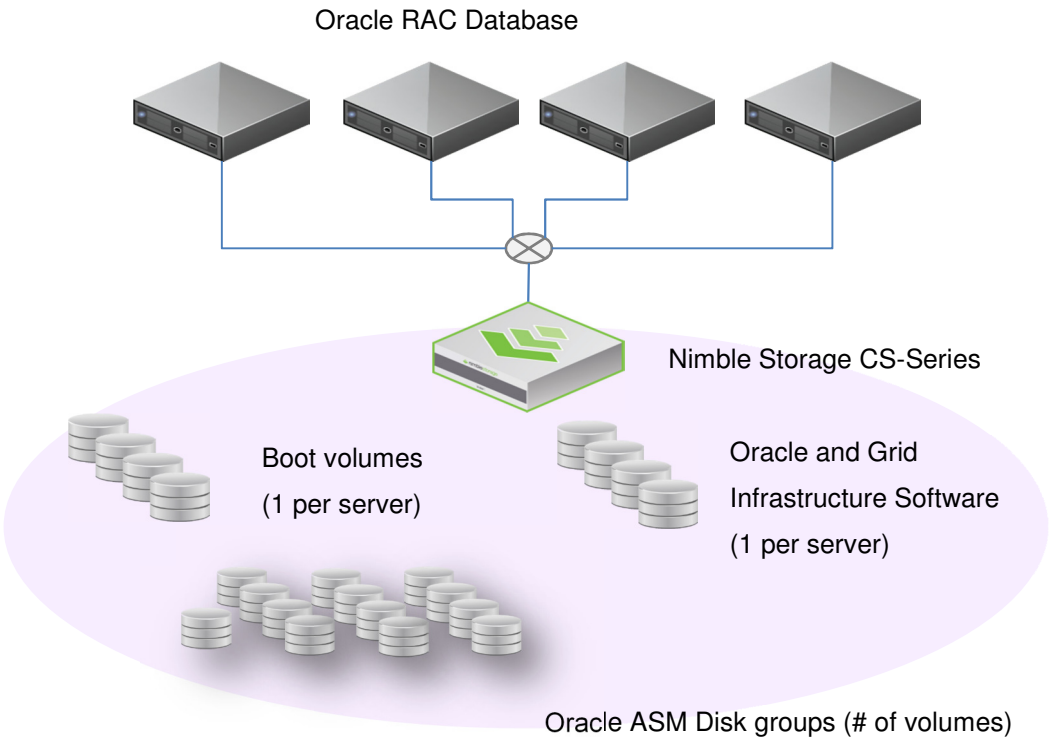
### *Zero-Copy Clones*

Nimble Storage arrays can create snapshot-based read/writeable clones of existing volumes instantly. These clones benefit from fast read and write performance, making them ideal for demanding applications such as VDI or database test/development.

### *InfoSight*

InfoSight leverages the power of deep-data analytics and cloud-based management to deliver true operational efficiency across all storage activities. It ensures the peak health of storage infrastructure by identifying problems, and offering solutions, in real time. InfoSight provides expert guidance for deploying the right balance of storage resources – dynamically and intelligently – to satisfy the changing demands of business-critical applications.

## *Oracle Database on Oracle Linux with Nimble Storage*




Oracle ASM Disk groups (# of volumes)

- OCR / Voting disk group (1)
- Data Disk group (8)
- Flash recovery area disk group (4)

When considering best practices for running Oracle databases including RAC on Oracle Linux, the areas to consider include performance, data protection and efficiency –especially as it related to test and development. This document covers the best practices including performance setting and volume setup with Oracle ASM.

## *Performance Settings*

When running Oracle database on Linux, there are many operating system settings that need to be tweaked to get the best performance and uptime. However, not all settings will make the Oracle database perform better. For an optimal performing database, there are many factors that need to be looked at. Such factors include, but not limited to:

- How the application was written to access the database data?
- Are the queries optimal?
- Are the logical database structures layout optimal for the workload (i.e. indexes, table partitioning)?
- What is the Server CPUs and memory profile?
- What type of IO Scheduler being used in Linux?
- What is the Queue depth setting?
- What File system is being used?
- What is the IO size chosen?
- How many Volumes/LUNs are created on storage?
- What is the number of IO paths to storage?

## *Fibre Channel Recommended Settings*

- Nimble OS should be at least 2.2.3
- 8Gb or 16Gb Brocade or Cisco MDS switches
- Dual fabric for HA
- Multipath
- 8Gb or 16Gb Qlogic or Emulex HBA
- Qlogic HBA Settings
    - qlport_down_retry = "0"
    - ql2xmaxqdepth = "32"

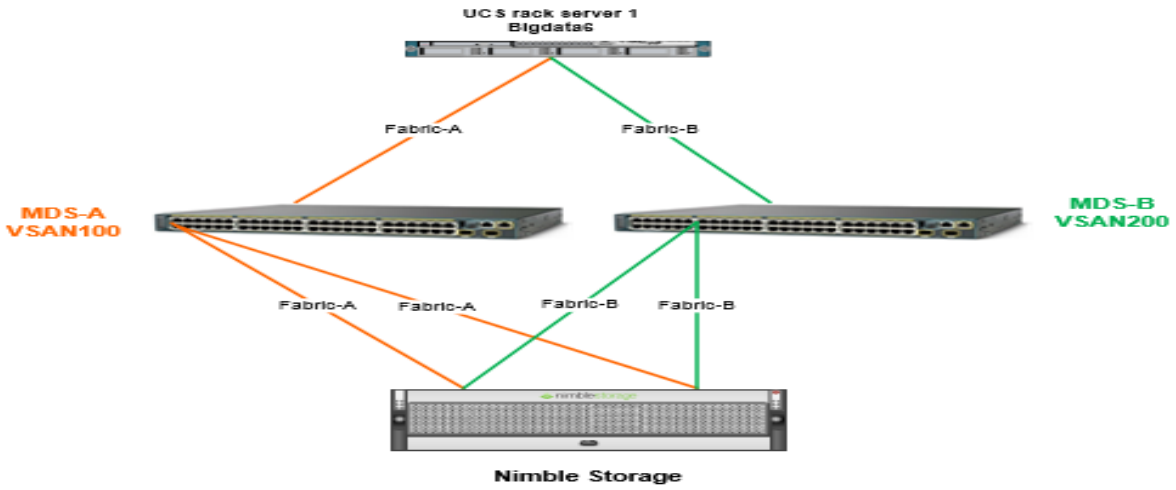



**Figure 1: Example of a Dual Fabric**

- **Multipath Settings for Fibre Channel**
  The multipath parameters in the */etc/multipath.conf* file should be set as follow in order to sustain a failover and performance. Nimble recommends the use of aliases for mapped LUNs.

```
defaults {
      user_friendly_names yes
      find_multipaths yes
}
devices {
  device {
       vendor                  "Nimble"
       product                 "Server"
       prio                    "alua"
       path_grouping_policy    group_by_prio
       path_checker            tur
       features                "1 queue_if_no_path"
       rr_weight               priorities
       rr_min_io_rq            1
       failback                immediate
       path_selector           "round-robin 0"
       dev_loss_tmo            infinity
       fast_io_fail_tmo        1
     }
}multipaths {
      multipath {
           wwid              20694551e4841f4386c9ce900dcc2bd34
           alias             ocr
     }
}
```

## *iSCSI Recommended Settings*

- Nimble OS should be at least 2.1.4
- Dual10GbE iSCSI Data Network Subnet
- iSCSI Timeout and Performance Settings
  Understanding the meaning of these iSCSI timeouts allows administrators to set these timeouts appropriately. These iSCSI timeouts parameters in the */etc/iscsi/iscsid.conf* file should be set as follow:

```
node.session.timeo.replacement_timeout = 120
node.conn[0].timeo.noop_out_interval = 5
node.conn[0].timeo.noop_out_timeout = 10
node.session.nr_sessions = 4
node.session.cmds_max = 2048
node.session.queue_depth = 1024
```

**= = = NOP-Out Interval/Timeout = = =**

node.conn[0].timeo.noop_out_timeout = [ value ]

iSCSI layer sends a NOP-Out request to each target. If a NOP-Out request times out (default - 10 seconds), the iSCSI layer responds by failing any running commands and instructing the SCSI layer to requeue those commands when possible. If dm-multipath is being used, the SCSI layer will fail those running commands and defer them to the multipath layer. The mulitpath layer then retries those commands on another path. If dm-multipath is not being used, those commands are retried five times (node.conn[0].timeo.noop_out_interval) before failing altogether.

node.conn[0].timeo.noop_out_interval [ value ]

Once set, the iSCSI layer will send a NOP-Out request to each target every [ interval value ] seconds.

**= = = SCSI Error Handler = = =**

If the SCSI Error Handler is running, running commands on a path will not be failed immediately when a NOP-Out request times out on that path. Instead, those commands will be failed after replacement_timeout seconds.

node.session.timeo.replacement_timeout = [ value ]

**Important**: Controls how long the iSCSI layer should wait for a timed-out path/session to reestablish itself before failing any commands on it. **The recommended setting of 120 seconds above allows ample time for controller failover**. Default is 120 seconds.

**Note**: If set to 120 seconds, IO will be queued for 2 minutes before it can resume.

The "***1 queue_if_no_path***" option in /etc/multipath.conf sets iSCSI timers to immediately defer commands to the multipath layer. This setting prevents IO errors from propagating to the application; because of this, you can set replacement_timeout to 60-120 seconds.

**Note**: Nimble Storage strongly recommends using dm-multipath for all volumes.

- **Multipath Settings for iSCSI**
  The multipath parameters in the */etc/multipath.conf* file should be set as follow in order to sustain a failover and performance. Nimble recommends the use of aliases for mapped LUNs.

```
defaults {
        user_friendly_names yes
        find_multipaths yes
}
devices {
    device {
        vendor               "Nimble"
        product              "Server"
        path_grouping_policy    group_by_serial
        path_selector          "round-robin 0"
        features              "1 queue_if_no_path"
        path_checker           tur
        rr_min_io_rq          10
        rr_weight             priorities
        failback              immediate
    }
}
multipaths {
        multipath {
                wwid                20694551e4841f4386c9ce900dcc2bd34
                alias               ocr
        }
}
```

- **iSCSI Data Network**
  Nimble recommends using 10GbE iSCSI for all databases.

2 separate subnets
2 x 10GbE iSCSI NICs
Use jumbo frames (MTU 9000) for iSCSI networks (Strongly Recommended)

Example of MTU setting for eth1:

```
DEVICE=eth1
HWADDR=00:25:B5:00:00:BE
TYPE=Ethernet
UUID=31bf296f-5d6a-4caf-8858-88887e883edc
ONBOOT=yes
NM_CONTROLLED=no
BOOTPROTO=static
IPADDR=172.18.127.134
NETMASK=255.255.255.0
MTU=9000
```

To change MTU on an already running interface:

```
[root@bigdata1 ~]# ifconfig eth1 mtu 9000
```

- **/etc/sysctl.conf**

```
net.core.wmem_max = 16780000
net.core.rmem_max = 16780000
net.ipv4.tcp_rmem = 10240 87380 16780000
net.ipv4.tcp_wmem = 10240 87380 16780000
```

Run sysctl –p command after editing the /etc/sysctl.conf file.

## *Linux Host Recommended Settings for both Fibre Channel and iSCSI*

- **max_sectors_kb**
  Change max_sectors_kb on all volumes to 1024 (default 512).

To change max_sectors_kb to 1024 for a single volume:

```
[root@bigdata1 ~]# echo 1024 > /sys/block/sd?/queue/max_sectors_kb
```

Change all volumes:

```
multipath -ll | grep sd  | awk -F":" '{print $4}' | awk '{print $2}' | while read LUN
do
  echo 1024 > /sys/block/${LUN}/queue/max_sectors_kb
done
```

**Note**: To make this change persistent after reboot, add the commands in /etc/rc.local file.

- **VM dirty writeback and expire**
  Change vm dirty writeback and expire to 100 (default 500 and 3000 respectively)

To change vm dirty writeback and expire:

[root@bigdata1 ~]# *echo 100 > /proc/sys/vm/dirty_writeback_centisecs*
[root@bigdata1 ~]# *echo 100 > /proc/sys/vm/dirty_expire_centisecs*

**Note**: To make this change persistent after reboot, add the commands in /etc/rc.local file.

- **CPU Scaling Governor**
  CPU Scaling Governor needs to be set at "*performance*"

To set the CPU scaling governor, run the below command.

[root@mktg04 ~]# *for a in $(ls -ld /sys/devices/system/cpu/cpu[0-9]* | awk '{print $NF}') ; do echo performance > $a/cpufreq/scaling_governor ; done*

**Note**: The setting above is not persistence after a reboot; hence the command needs to be executed when the server comes back online. To avoid running the command after a reboot, place the command in the /etc/rc.local file.

- **Disk IO Scheduler**
  IO Scheduler needs to be set at "*noop*"

To set IO Scheduler for all LUNs online, run the below command. **Note**: multipath must be setup first before running this command. Any additional LUNs added or server reboot will not automatically change to this parameter. Run the same command again if new LUNs are added or a server reboot.

[root@mktg04 ~]# *multipath -ll | grep sd | awk -F":" '{print $4}' | awk '{print $2}' | while read LUN; do echo noop > /sys/block/${LUN}/queue/scheduler ; done*

To set this parameter automatically, append the below syntax to */etc/grub.conf* file under the kernel line.

*elevator=noop*

## *Oracle OLTP with ASM*

### Recommended Nimble Volumes for Oracle ASM

Table1.

| Nimble Volume Role | Recommended Number of Volumes | Nimble Storage Caching Policy | Volume Block Size (Nimble Storage) |
|---|---|---|---|
| DATADG | 4 – Database server with 8 cores or less<br>8 – Database server with more than 16 cores | Yes – Normal | 8K |
| LOGDG | 4 | Yes – Normal | 4K |
| FRADG | 4 | No | 32K |

### Oracle Recommended Settings

Table2.

| Settings | Values |
|---|---|
| DB Block Size | 8KB |
| ASM Allocation Unit (AU) for diskgroups | 64MB |
| ASM Diskgroup Redundancy | External |
| # of db_writer_processes | 4 – DB server with 4 ASM disks for DATADG<br>8 – DB server with 8 ASM disks for DATADG |
| log_buffer size | ~1.6MB |
| _disk_sector_size_override | TRUE |
| Create online redo log files with block size | 4KB |
| filesystemio_options | setall |

Example of creating new log files:

*ALTER DATABASE ADD LOGFILE GROUP 5 ( '+LOGDG') SIZE 4096M BLOCKSIZE 4K;*
*ALTER DATABASE ADD LOGFILE GROUP 6 ( '+LOGDG') SIZE 4096M BLOCKSIZE 4K;*
*ALTER DATABASE ADD LOGFILE GROUP 7 ( '+LOGDG') SIZE 4096M BLOCKSIZE 4K;*
*ALTER DATABASE ADD LOGFILE GROUP 8 ( '+LOGDG') SIZE 4096M BLOCKSIZE 4K;*

## *Oracle OLTP with EXT4 File System*

### Recommended Nimble Volumes for Oracle with EXT4

Table3.

| Nimble Volume Role | Recommended Number of Volumes | Nimble Storage Caching Policy | Volume Block Size (Nimble Storage) |
|---|---|---|---|
| DATA LVM VG | 4 – Database server with 8 cores or less<br>8 – Database server with more than 16 cores | Yes – Normal | 8K |
| LOG LVM VG | 4 | Yes – Normal | 4K |
| FRA LVM VG | 4 | No | 32K |

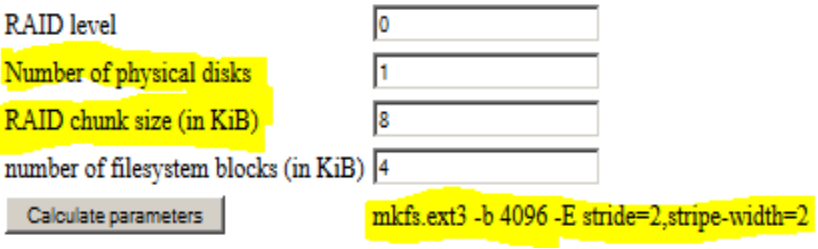
When creating an EXT file system on a logical volume, the ***stride*** and ***stripe-width*** options must be used.

For example:

*stride=2,stripe-width=16* (for Nimble performance policy 8KB block size with 8 volumes)
*stride=4,stripe-width=32* (for Nimble performance policy 16KB block size with 8 volumes)
*stride=8,stripe-width=64* (for Nimble performance policy 32KB block size with 8 volumes)

**Note**: The stripe-width value depends on the number of volumes, and the stride size. The calculator can be found here http://busybox.net/~aldot/mkfs_stride.html

For example: If there is one Nimble volume with 8KB block size performance policy, then it should look like this.




**Examples of LVM & EXT Setup:**

Create Volume Groups
[root@mktg04 ~]# *vgcreate vgextdata /dev/mapper/extdata[1-8]*
[root@mktg04 ~]# *vgcreate vgextlog /dev/mapper/extlog[1-4]*
[root@mktg04 ~]# *vgcreate vgextarch /dev/mapper/extarch[1-4]*

Create Logical Volume
[root@mktg04 ~]# *lvcreate -l <# of extents> -i 8 -I 4096 -n vol1 vgextdata*
[root@mktg04 ~]# *lvcreate -l <# of extents> -i 4 -I 4096 -n vol1 vgextlog*
[root@mktg04 ~]# *lvcreate -l <# of extents> -i 4 -I 4096 -n vol1 vgextarch*

Create EXT file system
[root@mktg04 ~]# *mkfs.ext4 /dev/vgextdata/vol1 -b 4096 -E stride=2,stripe-width=16*
[root@mktg04 ~]# *mkfs.ext4 /dev/vgextlog/vol1 -b 4096*
[root@mktg04 ~]# *mkfs.ext4 /dev/vgextarch/vol1 -b 4096 -E stride=8,stripe-width=32*

Mount options in /etc/fstab file for iSCSI
*/dev/vgextdata/vol1 /u01/app/extdata ext4 _netdev,noatime,nodiratime,discard,barrier=0 0 0*
*/dev/vgextlog/vol1 /u01/app/extlog ext4 _netdev,noatime,nodiratime,discard,barrier=0 0 0*
*/dev/vgextarch/vol1 /u01/app/extarch ext4 _netdev,noatime,nodiratime,discard,barrier=0 0 0*

Mount options in /etc/fstab file for Fibre Channel
*/dev/vgextdata/vol1 /u01/app/extdata ext4 noatime,nodiratime,discard,barrier=0 0 0*
*/dev/vgextlog/vol1 /u01/app/extlog ext4 noatime,nodiratime,discard,barrier=0 0 0*
*/dev/vgextarch/vol1 /u01/app/extarch ext4 noatime,nodiratime,discard,barrier=0 0 0*

## Oracle Recommended Settings

Table4.

| Settings | Values |
|---|---|
| DB Block Size | 8KB |

| | |
|---|---|
| # of db_writer_processes | 4 – DB server with 4 ASM disks for DATADG<br>8 – DB server with 8 ASM disks for DATADG |
| log_buffer size | ~1.6MB |
| _disk_sector_size_override | TRUE |
| Create online redo log files with block size | 4KB |
| filesystemio_options | setall |

Example of creating new log files:

*ALTER DATABASE ADD LOGFILE GROUP 5 ( '/u01/app/extlog/log5') SIZE 4096M BLOCKSIZE 4K;*
*ALTER DATABASE ADD LOGFILE GROUP 6 ( '/u01/app/extlog/log6') SIZE 4096M BLOCKSIZE 4K;*
*ALTER DATABASE ADD LOGFILE GROUP 7 ( '/u01/app/extlog/log7') SIZE 4096M BLOCKSIZE 4K;*
*ALTER DATABASE ADD LOGFILE GROUP 8 ( '/u01/app/extlog/log8') SIZE 4096M BLOCKSIZE 4K;*

## *Oracle DSS with ASM*

### Recommended Nimble Volumes for Oracle ASM

Table5.

| Nimble Volume Role | Recommended Number of Volumes | Nimble Storage Caching Policy | Volume Block Size (Nimble Storage) |
|---|---|---|---|
| DATADG | 4 – Database server with 8 cores or less<br>8 – Database server with more than 16 cores | Yes – Normal | 32K |
| LOGDG | 4 | Yes – Normal | 4K |
| FRADG | 4 | No | 32K |

### Oracle Recommended Settings

Table6.

| Settings | Values |
|---|---|
| DB Block Size | 32KB |
| ASM Allocation Unit (AU) for diskgroups | 64MB |
| ASM Diskgroup Redundancy | External |
| # of db_writer_processes | 4 – DB server with 4 ASM disks for DATADG<br>8 – DB server with 8 ASM disks for DATADG |
| log_buffer size | ~1.6MB |
| _disk_sector_size_override | TRUE |
| Create online redo log files with block size | 4KB |
| filesystemio_options | setall |

Example of creating new log files:

*ALTER DATABASE ADD LOGFILE GROUP 5 ( '+LOGDG') SIZE 4096M BLOCKSIZE 4K;*
*ALTER DATABASE ADD LOGFILE GROUP 6 ( '+LOGDG') SIZE 4096M BLOCKSIZE 4K;*
*ALTER DATABASE ADD LOGFILE GROUP 7 ( '+LOGDG') SIZE 4096M BLOCKSIZE 4K;*
*ALTER DATABASE ADD LOGFILE GROUP 8 ( '+LOGDG') SIZE 4096M BLOCKSIZE 4K;*

## *Oracle DSS with EXT4 File System*

### Recommended Nimble Volumes for Oracle with EXT4

Table7.

| Nimble Volume Role | Recommended Number of Volumes | Nimble Storage Caching Policy | Volume Block Size (Nimble Storage) |
|---|---|---|---|

| | | | |
|---|---|---|---|
| DATA LVM VG | 4 – Database server with 8 cores or less<br>8 – Database server with more than 16 cores | Yes – Normal | 32K |
| LOG LVM VG | 4 | Yes – Normal | 4K |
| FRA LVM VG | 4 | No | 32K |

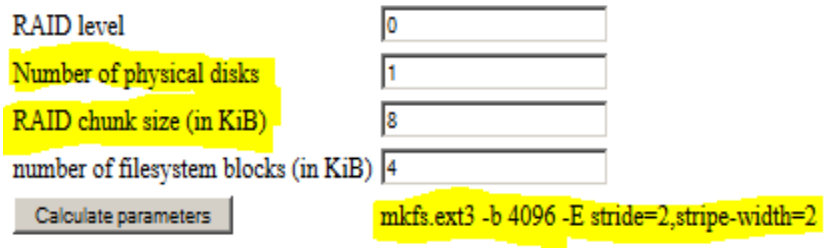
When creating an EXT file system on a logical volume, the ***stride*** and ***stripe-width*** options must be used.

For example:

*stride=2,stripe-width=16* (for Nimble performance policy 8KB block size with 8 volumes)
*stride=4,stripe-width=32* (for Nimble performance policy 16KB block size with 8 volumes)
*stride=8,stripe-width=64* (for Nimble performance policy 32KB block size with 8 volumes)

**Note**: The stripe-width value depends on the number of volumes, and the stride size. The calculator can be found here http://busybox.net/~aldot/mkfs_stride.html

For example: If there is one Nimble volume with 8KB block size performance policy, then it should look like this.




**Examples of LVM & EXT Setup:**

Create Volume Groups
[root@mktg04 ~]# *vgcreate vgextdata /dev/mapper/extdata[1-8]*
[root@mktg04 ~]# *vgcreate vgextlog /dev/mapper/extlog[1-4]*
[root@mktg04 ~]# *vgcreate vgextarch /dev/mapper/extarch[1-4]*

Create Logical Volume
[root@mktg04 ~]# *lvcreate -l <# of extents> -i 8 -I 4096 -n vol1 vgextdata*
[root@mktg04 ~]# *lvcreate -l <# of extents> -i 4 -I 4096 -n vol1 vgextlog*
[root@mktg04 ~]# *lvcreate -l <# of extents> -i 4 -I 4096 -n vol1 vgextarch*

Create EXT file system

[root@mktg04 ~]# *mkfs.ext4 /dev/vgextdata/vol1 -b 4096 -E stride=2,stripe-width=16*
[root@mktg04 ~]# *mkfs.ext4 /dev/vgextlog/vol1 -b 4096*
[root@mktg04 ~]# *mkfs.ext4 /dev/vgextarch/vol1 -b 4096 -E stride=8,stripe-width=32*

Mount options in /etc/fstab file for iSCSI

```
/dev/vgextdata/vol1   /u01/app/extdata     ext4   _netdev,noatime,nodiratime,discard,barrier=0   0 0
/dev/vgextlog/vol1    /u01/app/extlog      ext4   _netdev,noatime,nodiratime,discard,barrier=0   0 0
/dev/vgextarch/vol1   /u01/app/extarch     ext4   _netdev,noatime,nodiratime,discard,barrier=0   0 0
```

Mount options in /etc/fstab file for Fibre Channel

```
/dev/vgextdata/vol1   /u01/app/extdata     ext4   noatime,nodiratime,discard,barrier=0   0 0
/dev/vgextlog/vol1    /u01/app/extlog      ext4   noatime,nodiratime,discard,barrier=0   0 0
/dev/vgextarch/vol1   /u01/app/extarch     ext4   noatime,nodiratime,discard,barrier=0   0 0
```

## Oracle Recommended Settings

Table8.

| Settings | Values |
|---|---|
| DB Block Size | 32KB |
| # of db_writer_processes | 4 – DB server with 4 ASM disks for DATADG<br>8 – DB server with 8 ASM disks for DATADG |
| log_buffer size | ~1.6MB |
| _disk_sector_size_override | TRUE |
| Create online redo log files with block size | 4KB |
| filesystemio_options | setall |

Example of creating new log files:

```
ALTER DATABASE ADD LOGFILE GROUP 5 ('/u01/app/extlog/log5') SIZE 4096M BLOCKSIZE 4K;
ALTER DATABASE ADD LOGFILE GROUP 6 ('/u01/app/extlog/log6') SIZE 4096M BLOCKSIZE 4K;
ALTER DATABASE ADD LOGFILE GROUP 7 ('/u01/app/extlog/log7') SIZE 4096M BLOCKSIZE 4K;
ALTER DATABASE ADD LOGFILE GROUP 8 ('/u01/app/extlog/log8') SIZE 4096M BLOCKSIZE 4K;
```

**Nimble Storage, Inc.**

211 River Oaks Parkway, San Jose, CA  95134

Tel: 877-364-6253)  |  www.nimblestorage.com  |  info@nimblestorage.com