TEXAS MEMORY SYSTEMS, AN IBM COMPANY

RAMSAN FLASH SYSTEMS

INTEGRATION GUIDE

Version 3.0

Table of Contents

Preface

Document Overview

This document details how to get the maximum performance from the RamSan Flash Systems. This guide will cover the best practices for redundancy as well as performance in single-server and multi-server environments.

Revision History

The following table describes revisions to this document:

Version	Comments	Release Date
1.0	Initial release.	NA
2.0	Added OS-specific chapters.	NA
2.1	Added dmp_failed_io_threshold.	NA
2.2	Added Linux InfiniBand Integration section, and various edits.	NA
2.3	Added AIX, HP-UX, and Storage Mode chapters; added note that Veritas vxdisk command only applies to Windows; unified page layout.	
2.4	Added further AIX details -- ACA note, ODM installation instructions, and appendix.	NA
2.5	Fixed vxdisk comment to correctly say that it is to be used only in Unix (not available in Windows).	NA
2.6	Initial release of IB Controller and HCA sections. Integrated IB and HCA into overall guide. Initial release of Oracle, MS SQL Server, and Sybase sections. Added information on RamSan-640, RamSan-710, and RamSan-810.	NA
2.7	Added VSR® registered trademark symbol and deleted ™.	NA
2.8	Reverted change made in 2.5 - vxdisk command refers to Windows only, not Unix.	NA
2.9	Updated internal document references and added Index.	NA
2.10	Added note for HP-UX 11.11 to set Loop ID Assignment to HARD to maintain consistent device paths across reboots.	Unreleased
3.0	Initial integration of the new RS-720/820 systems into document.	2012/11/09

Chapter 1 – Introduction

Texas Memory Systems, an IBM Company designed its Solid State Disk (SSD) products using standard interfaces that support a wide range of servers. Servers are generally configured for Hard Disk Drives (HDD) and some tweaks can be made to better take advantage of the speed of SSDs.

This document goes through multipathing best practices, RamSan Flash System storage configurations, and alignment strategies to reach the optimal performance from the RamSan Flash Systems. For the purpose of this guide, the RamSan Flash Systems series includes the:

- RamSan-620
- RamSan-625
- RamSan-630
- RamSan-640
- RamSan-710
- RamSan-720
- RamSan-810
- RamSan-820

Additional information for these systems, and any other RamSan systems, can be obtained by contacting support@ramsan.com.

Chapter 2 – Connectivity Best Practices

Connecting the RamSan follows traditional storage appliance practices and can allow for point-to-point, fabric-attached, and mixed architectures. This chapter covers these practices as well as optimal cabling for highest performance and redundancy. In this chapter, the performance discussed assumes that the maximum Flash supported by a system is installed.

2.1 RamSan Interface Controllers

The RamSan Flash Systems series includes several different interface controller options, including 4 Gb Fibre Channel controllers, 8 Gb Fibre Channel controllers, and QDR (40 Gb) InfiniBand controllers. See *Table 1* for a summary of the options by system.

System	4 Gb FC	8 Gb FC	QDR (40 Gb) IB	Max. Number of Controllers (Ports)
RamSan-620	X			4 (8)
RamSan-625	X			4 (8)
RamSan-630	X	X	X	5 (10)
RamSan-640	X	X	X	5 (10)
RamSan-710		X	X	2 (4)
RamSan-720		X	X	2 (4)
RamSan-810		X	X	2 (4)
RamSan-820		X	X	2 (4)

Table 1: Interface Controller Options by System

To increase the bandwidth and IOPS performance limits of the RamSan Flash Systems series, add more interface controllers. *Table 1* also lists the maximum number of interface controllers and ports by system.

2.1.1 RamSan FC Controllers

Each dual-ported 4 Gb or 8 Gb Fibre Channel controller in a RamSan Flash Systems provides bandwidth of close to 800 MB/s or 1,600 MB/s.

- When utilizing the full four 4 Gb FC controllers in a RamSan-620 or RamSan-625, there is an aggregate bandwidth of 3 GB/s of read or write bandwidth.
- When utilizing the full five 8 Gb FC controllers on the RamSan-630 or RamSan-640, there is an aggregate bandwidth of 7.5 GB/s of read or write bandwidth.
- When using the two 8 Gb FC controllers in the RamSan-710/810 or RamSan-720/820, there is an aggregate bandwidth of 3.2 GB/s.

A single optical transceiver is the most likely component to experience failure on an FC controller. There are two hot-pluggable transceivers on a FC controller so a single transceiver does not fault the entire card. However, both ports do share a common processor. To eliminate any FC failure point, multipathing is recommended to span across two or more controllers before utilizing the second ports on the controllers. The IOPS and bandwidth limits of the RamSan Flash Systems increase with the number of controllers.

For optimal performance, connect the A-ports of each FC before utilizing the B-ports. Therefore, FC-1A through FC-5A should be connected before FC-1B through FC-5B.

The RamSan Flash Systems allows very flexible presentation of LUNs through its FC controllers. A LUN can be presented and accessed through every port simultaneously. The loss of one controller does not affect the I/O on another controller. This is commonly referred to as supporting Active/Active or Active/Active symmetric multipathing.

Almost all major operating systems incorporate a multipathing option. For further questions about multipathing not covered in this document, please contact RamSan support at support@ramsan.com.

2.1.2 RamSan IB Controllers

Most RamSan Flash Systems support InfiniBand (IB) controllers (the only exceptions are the RamSan-620 and -625). Each dual-ported QDR IB controller provides bandwidth of close to 3 GB/s.

- The RamSan-630 can provide an aggregate read or write bandwidth of 10 GB/s across four or more IB controllers.
- The RamSan-640 can provide an aggregate read or write bandwidth of 8 GB/s across three or more IB controllers.
- The RamSan-710/810 or RamSan-720/820 can provide an aggregate read or write bandwidth of 5 GB/s across both IB controllers.

For optimal performance, connect the A-ports of each IB before utilizing the B-ports. Therefore, IB-1A through IB-5A should be connected before IB-1B through IB-5B.

The RamSan Flash Systems allows very flexible presentation of LUNs through its IB controllers. A LUN can be presented and accessed through every port simultaneously. The loss of one controller does not affect the I/O on another controller. This is commonly referred to as supporting Active/Active or Active/Active symmetric multipathing.

Almost all major operating systems incorporate a multipathing option. For further questions about multipathing not covered in this document, please contact RamSan support at support@ramsan.com.

2.2 Host Bus Adapter (HBA)

Texas Memory Systems, an IBM Company recommends using dual-ported Host Bus Adapters (HBAs) when possible. Dual-ported and quad-ported HBAs provide additional ports for aggregating bandwidth while conserving the limited expansion slots on most servers. In an ideal case, multiple dual-ported HBAs are used within each server for redundancy. RamSans are tested for interoperability against all major HBA vendors. For any questions about support for a particular HBA, please contact RamSan support at support@ramsan.com.

2.3 Host Channel Adapter (HCA)

Texas Memory Systems, an IBM Company recommends using dual-ported Host Channel Adapters (HCAs) when possible. Dual-ported HCAs provide additional ports for aggregating bandwidth while conserving the limited expansion slots on most servers. In an ideal case, multiple dual-ported HCAs are used within each server for redundancy.

Texas Memory Systems, an IBM Company also recommends Quad Data Rate (QDR) HCAs to maximize the bandwidth to the RamSan-630, -640, -710/810, and -720/820. Most QDR HCAs utilize PCI-Express 2.0 instead of 1.0 to allow for the higher rated bandwidth. Texas Memory Systems, an IBM Company recommends using servers with PCI-Express 2.0 or higher expansion slots for the highest bandwidth needs.

RamSans are tested for interoperability against all major HCA vendors. For the highest performance requirements, Texas Memory Systems, an IBM Company recommends Mellanox HCAs. For any questions about support for a particular HCA, please contact RamSan support at support@ramsan.com.

2.4 Cabling

Cabling should be designed to provide resiliency across RamSan ports, FC/IB controllers, switches, server HBA/HCA ports, and HBAs/HCAs. All high availability (HA) concepts of a dual-fabric setup apply to the RamSan.

When connecting to redundant fabrics, connect the A-ports to one fabric and the B-ports to the other fabric.

One key element to recognize when cabling a RamSan is the use of available paths. In order to take advantage of RamSan ports, there must be an equal number of server ports. Otherwise, there are underutilized ports on one side of the fabric.

The RamSan is built to deliver I/O through all connected ports.

- The RamSan-620 and RamSan-625 support up to 4 FC controllers and a total of 8 available ports.

- The RamSan-630 and RamSan-640 support up to 5 FC/IB controllers and a total of 10 available ports.
- The RamSan-710/810 and RamSan-720/820 support up to 2 FC/IB controllers and a total of 4 available ports.

Using all ports is recommended if the application can benefit from additional bandwidth.

2.5 Zoning

In a switched fabric deployment, it is a common practice to isolate one application's Server-Storage devices from other applications' Server-Storage devices to prevent cross traffic and help ease maintenance. TMS recommends zoning in all multi-server environments.

Zoning is best deployed in a 1:1 port ratio with 1 server's HBA/HCA port for 1 RamSan FC/IB port. A server HBA/HCA port serving I/O to multiple FC/IB ports within the same RamSan results in excessive paths since the server's single HBA/HCA port is the limiting factor for performance.

Note:

When connecting the RamSan to a split-fabric architecture, be sure each zoned-server utilizes multiple FCs/IBs.

- *For the RamSan-620 and -625, a good practice is to access ports FC-1A through FC-4A, and then **reverse** the remaining ports from FC-4B through FC-1B. This way, the first server sees at least ports FC-**1**A and FC-**4**B and does not depend upon a single FC for availability.*
- *For the RamSan-630 and RamSan-640, access ports FC-1A through FC-5A (or IB-1A through IB-5A), and then **reverse** the remaining ports from FC-5B through FC-1B (or IB-5B through IB-1B). This way, the first server sees at least ports FC-**1**A and FC-**5**B and does not depend upon a single FC for availability.*
- *For the RamSan-710/810 and RamSan-720/820, access ports FC-1A through FC-2A (or IB-1A through IB-2A), and then **reverse** the remaining ports from FC-2B through FC-1B (or IB-2B through IB-1B). This way, the first server sees at least ports FC-**1**A and FC-**2**B and does not depend upon a single FC for availability.*

2.6 LUN Masking (Licensed Feature)

LUN Masking is a utility that allows separate servers to access separate LUNs through a common controller port. It is possible to use LUN Masking in place of zoning, but it is better utilized as a supplement to zoning. If multiple servers access different LUNs through a common RamSan port, then LUN Masking can be used to present only the LUNs needed by the server.

LUN Masking requires a license on the RamSan Flash Systems.

Chapter 3 – Sector Sizes

RamSan Flash Systems have the option of presenting each LUN with a unique sector size. These sizes range from a default of 512 byte up to a high-performance 4096 byte (4 KB) sector size. Logical Units configured with the 4 KB sector size will provide optimal performance as long as the application and underlying operation system are compatible with the advanced sizing.

Most file systems can take advantage of 4 KB sectors but the hosted application must also support the sector size.

Note:

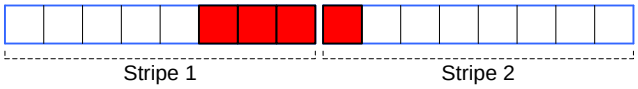
All file systems on the RamSan Flash Systems should be formatted at, or at a multiple of, 4 KB. For example, file systems formatted at 8 KB or 64 KB allocation sizes are satisfactory as they are a multiple of 4 KB.

Sector size changes are permitted in the RamSan GUI and CLI, but are destructive to any existing filesystem on the LUN. Therefore, we recommend choosing the sector size before a production deployment.

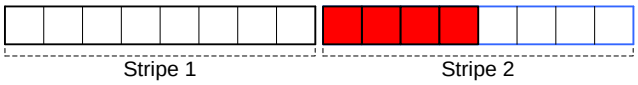
For more information on applications supporting 4 KB sectors, please contact support@ramsan.com.

Chapter 4 – Alignment

File system alignment is a technique to match the file system I/O requests with important block boundaries in the physical storage system. This is important in any system that implements a RAID layout, as I/O requests that fall within the bounds of a single stripe will have better performance than an I/O request that crosses over multiple stripes. When an I/O occurs across the end-point of one stripe and into another, the controller must modify both stripes to maintain their consistency. This is visually depicted in the following graphic:



When the I/O is aligned, the result is a single modified stripe.



Note:

For the RamSan Flash Systems with a 512 byte sector size, alignment is optimal at 4 KB. Unaligned accesses, those that start at a non-4K divisible address or are not a multiple of 4 KB in size, are serviced at much higher response times and can also significantly reduce the performance of aligned accesses that were issued in parallel.

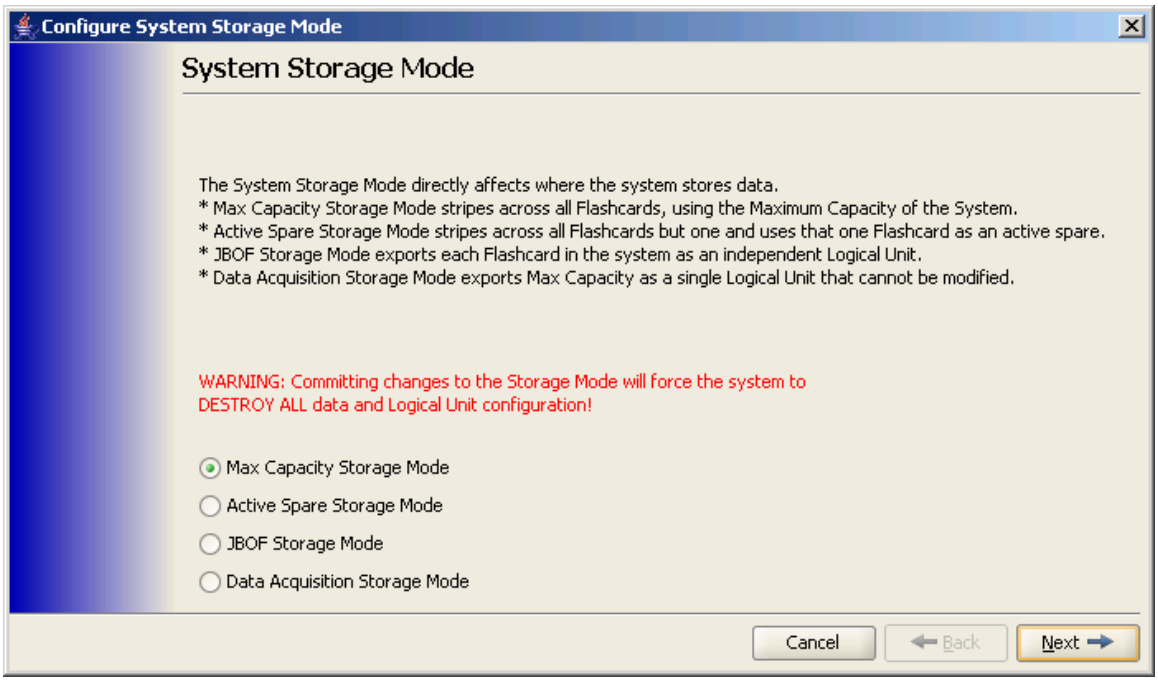
When using a 4 KB sector size on a LUN, there is no need to force alignment at the host or application.

See the Appendix for details on confirming that an application is submitting 4K aligned IO.

I/O Alignment is usually visible only through writes to the disk. If an application is seeing higher-than-expected response times for the write activity, contact support@ramsan.com to help determine the optimal configuration for the application.

Chapter 5 – Storage Modes

RamSan Flash Systems have adjustable Storage Modes that increase flexibility and resiliency where appropriate. The available Storage Modes are Max Capacity, Active Spare, JBOF, and Data Acquisition. Setting the Storage Mode alters the internal placement of data within the RamSan and is not a data in place operation.



Active Spare and JBOF Storage Modes are licensed storage modes.

5.1 Max Capacity Storage Mode

Max Capacity presents the maximum usable capacity for creating LUNs. In this mode, 240 GiB of capacity is presented for each Flash Module in an aggregate pool of capacity. LUNs are created across all available Flash Modules to achieve the highest performance. Note that in this mode, if maintenance is performed that requires the replacement of a Flash card in the RamSan Flash Systems all of the data will need to be restored from a backup after the maintenance. This mode is recommended for deployments where RamSan Flash Systems are mirrored or preferred read mirrored to other storage.

5.2 Active Spare Storage Mode

Active Spare reserves the last available card in the system as a spare for resiliency and serviceability. The most probable element of failure in the RamSan Flash Systems is a Flash chip, which is protected against failure

by the Variable Stripe RAID® (VSR®) or RAID-5 layout of the Flash within each Flash card. In the event of a Flash chip failure in the RamSan Flash Systems, Active Spare migrates the data from the Flash card with a degraded RAID on to the designated spare to return to full RAID protection.

The failover process is a data migration process that takes a few hours of non-blocking activity. This means the application can still operate under normal load while the data is copied to the replacement module. Once the module has finished its failover, a maintenance window should be conveniently scheduled to replace the original module.

In this storage mode there are still a few components that are not protected by Active Spare, so there is a risk of an unrecoverable component failure. If a deployment requires no single point of failure, mirrored RamSan Flash Systems using the Max Capacity Storage mode is the recommended solution.

5.3 JBOF Storage Mode

JBOF, or Just a Bunch of Flash, is synonymous with the industry JBOD. This mode presents each Flash board out as its own LUN. This is primarily meant to be used in conjunction with host-level storage protection, such as a software RAID or Oracle's ASM.

5.4 Data Acquisition Storage Mode

This storage mode is specific to specialized Data Acquisition applications and offers an increase in write performance. This should only be used for write-only workloads that benefit from increased bandwidth after consultation with Texas Memory Systems, an IBM Company support.

Chapter 6 – Microsoft Windows

The RamSan will see the most benefit from server-class operating systems where multipathing and logical volumes are more feature rich. There are some applications that are dependent upon the workstation class OS and can utilize the RamSan performance.

6.1 Directly-Attached FC

Arbitrated Loop (AL) or Point-to-Point (PP) topology may be used on the RamSan FC configuration for Windows hosts.

6.2 Sector Size Support

Windows Server 2003 and Windows Server 2008 both support the large 4 KB sectors and have very good multipathing solutions. On the desktop series of Windows, multipathing is not available and 4 KB sector support is limited to Windows Vista and Windows 7.

Most Microsoft applications will also support 4 KB sector support. SQL Server and Exchange will support 4 KB, but require a migration procedure when using an existing dataset. Contact support@ramsan.com to see if a specific application supports large 4 KB sectors.

When using LUNs with a 512 byte sector size, alignment is critical on Windows Server 2003 and XP.

6.3 Alignment for Windows 2000/XP and Windows Server 2000/2003

Prior to Vista and Server 2008, Windows will offset the partition by 63 sectors, or 31.5KB. To align to the preferred 4 KB, the offset must be done using the *diskpart.exe* utility.

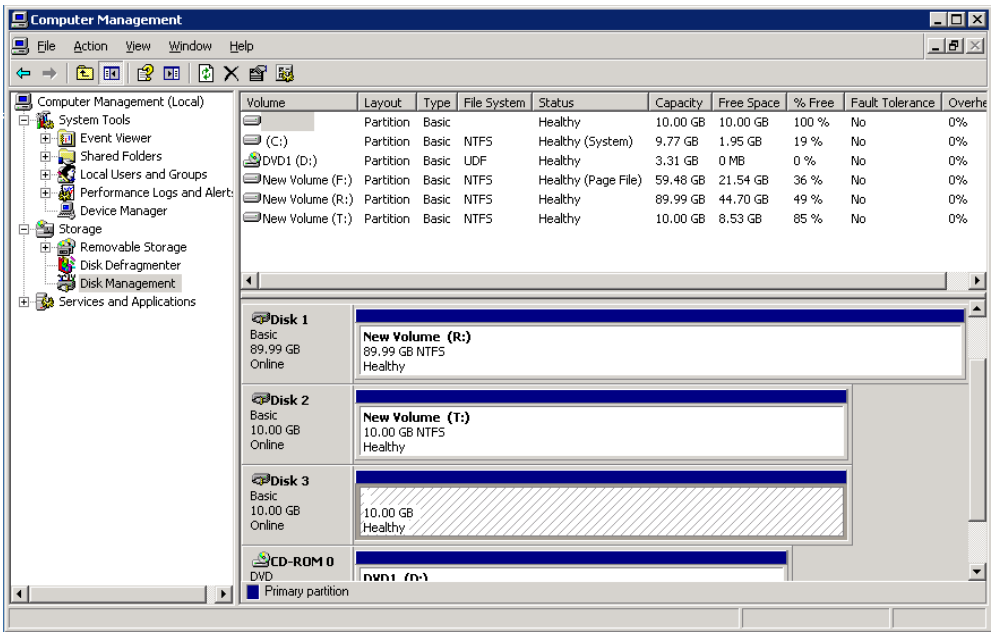
First, select the disk (LUN) that will hold the file system:

```
DISKPART> list disk

  Disk ###  Status      Size     Free     Dyn  Gpt
  --------  ----------  -------  -------  ---  ---
  Disk 0    Online        69 GB      0 B
  Disk 1    Online        90 GB      0 B
  Disk 2    Online        10 GB      0 B
  Disk 3    Online        10 GB    10 GB
DISKPART> select disk 3

Disk 3 is now the selected disk.
```

Once selected, a partition can be made on the disk.

```
DISKPART> create partition primary align=64

DiskPart succeeded in creating the specified partition.
```

Now, a file system or drive letter (RAW access) can be assigned to the aligned partition for application use. These can be done to the same Disk 3 through the Disk Management MMC.



Windows Vista, Server 2008, and later OSes default to 1024 KB offsets which align to the RamSan's optimal size.

If installing the Windows XP/Server 2003 OS on the RamSan, it is recommended that you create the partition on the LUN prior to installation. This can be done relatively easily from a Linux LiveCD or by presenting the LUN to another Windows host and disconnecting after the partitioning.

6.4 Windows Server 2003 Multipathing

Windows Server 2003 has a built-in MPIO driver provided by Microsoft. This driver is responsible for aggregating the links of RamSan systems and reporting the addition or removal of links while online to the kernel. In order to utilize this feature with a RamSan, a Device Specific Module (DSM) is provided by Texas Memory Systems, an IBM Company to alert the MPIO driver of the RamSan's identity and its Active/Active capabilities. This driver is available free upon request to support@ramsan.com.

If running EMC PowerPath, it must be upgraded to 4.6 or later before using the RamSan DSM. All major storage vendors have support for the MPIO multipathing and coexist safely due to the common driver in Windows.

6.5 Windows Server 2008/2012 Multipathing

Windows Server 2008/2012 no longer requires a DSM provided by Texas Memory Systems, an IBM Company. Instead, the Multipath Feature option must be installed on the Windows 2008/2012 server.

Installation instructions for the MPIO feature can be followed from Microsoft's TechNet:

http://technet.microsoft.com/en-us/library/ee619752(WS.10).aspx

After the MPIO Feature has been installed, the Load Balance Policy should be set on all RamSan LUNs to *Lowest Queued Path*. All available paths to the RamSan LUNs are then used to aggregate bandwidth. The Load Balance Policy is set through the Properties Dialog of the Multipath Disk Device in Device Manager.

6.6 Disk Retry Time Out

Windows has a default Disk Command *TimeOutValue* of 60 seconds. Therefore, if a SCSI command does not complete due to a dropped frame, CRC Error, or reset link, the application must wait 60 seconds before the I/O is retried. Most applications will complain for I/Os longer than 30 seconds and sometimes less.

It is recommended by TMS and Microsoft to adjust the Disk *TimeOutValue* to a lower value in registry key:

HKLM\SYSTEM\CurrentControlSet\services\Disk

Reference: http://technet.microsoft.com/en-us/library/aa997069(EXCHG.80).aspx

For a **non-clustered** disk, set the *TimeOutValue* to a decimal value of **10**. For a **clustered** disk, set the *TimeOutValue* to a decimal value of **20**.

Chapter 7 – Linux

Most distributions of Linux will have the same configuration and optimizations. Questions regarding distro or kernel-specific options can be directed to support@ramsan.com.

7.1 Directly-Attached FC

Arbitrated Loop (AL) or Point-to-Point (PP) topology may be used on the RamSan FC configuration for Linux hosts.

7.2 Sector Size Support

Linux will support large 4 KB sectors and will also do a translation for most applications that still depend upon 512 byte sectors. This means that applications that are not sector-size aware can utilize the 4 KB sector performance while the file system will handle the small-block writes that cannot be done to 4 KB sectors with Direct IO.

For clustered applications where DirectIO should be implemented, the application should be presented with 512 byte sector storage and aligned at best effort.

7.3 Alignment

Linux defaults to a 63-sector offset. To align a partition in Linux, use fdisk and do the following:

```
# fdisk /dev/mapper/ramsan

Command (m for help): u
Changing display/entry units to sectors

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First sector (63-16777215, default 63): 128
Last sector or +size or +sizeM or +sizeK (128-16777215,
default 16777215): <enter>
Using default value 16777215

Command (m for help): w
The partition table has been altered!
```

The newly created partition now has an offset of 64 KB and will perform optimally with an aligned application.

If installing the Linux OS on the RamSan, create the partition scheme prior to the installation process. For most Linux distributions, this requires booting to the text-based installer and switching consoles (Alt+F2) to the command prompt before continuing through the installer.

7.4 Multipathing

Linux kernels on 2.6 and higher support multipathing through *device-mapper-multipath*. This package can coexist with other multipathing solutions as long as the other storage is excluded from *device-mapper*. Appendix A includes a template for the multipath.conf file for utilizing the RamSan to its full potential.

Since the RamSan controllers provide true Active/Active I/O, the rr_min_io field in the multipath.conf file is set to 1. This results in a round-robin distribution of I/O across all available paths. If the I/O is more sequential in nature, then the rr_min_io field should be slightly increased by factors of 2 for a slight performance gain if using buffered I/O (non-DIRECT).

For 4 KB Sectors support, the PathSelector option in the multipath.conf must be set to "tur" for Test Unit Ready.

7.5 Optimizations

To further improve the performance on Linux, append the kernel parameter "elevator=noop" to disable the I/O scheduler. This will help reduce latency on small-block requests to the RamSan. This will also greatly improve performance of mixed reads and writes to the file system.

```
Example entry in /boot/grub/menu.lst (/etc/grub.conf):

title Red Hat Enterprise Linux Server (2.6.18-164.el5)
        root (hd0,0)
        kernel /vmlinuz-2.6.18-164.el5 ro root=LABEL=/
elevator=noop rhgb quiet
        initrd /initrd-2.6.18-164.el5.img
```

7.6 InfiniBand Integration

7.6.1 Integration

To integrate with InfiniBand (IB), the RamSan provides block storage via the SCSI RDMA Protocol (SRP).

Linux requires several software modules to connect to the RamSan through IB and SRP. Namely, make sure to install the *srp* and *srptools* modules, and make sure to install drivers for the server's Host Channel Adapter (HCA). It is recommended to use the OpenFabrics Enterprise Distribution (OFED) package from http://www.openfabrics.org to install these modules either individually or using the "Install All" option.

In order for SRP to work with the RamSan, make sure the following options in /etc/infiniband/openib.conf are set:

```
# Load SRP module
SRP_LOAD=yes

# Enable SRP High Availability daemon
SRPHA_ENABLE=yes
SRP_DAEMON_ENABLE=yes
```

These settings will cause SRP and the SRP daemon to load automatically when the IB driver boots. The SRP daemon will automatically discover and connect to IB SRP disks.

The *SRPHA_ENABLE=yes* setting is a best practice. This triggers the multipath daemon to create a multipath disk target when a new disk is detected, and works in conjunction with the multipath best practices given in *Section 7.4 Multipathing*.

Finally, using IB requires a Subnet Manager (SM). An existing IB network will already have an SM. In many cases, an IB switch acts as the SM. If an SM is needed, install OpenSM (included with OFED) and start it on a single server in the network by executing:

```
# /etc/init.d/opensmd start
```

Note that this script only opens an SM on a single port. If there are multiple ports directly connected to the RamSan, a custom script will be needed to start the SM on all ports.

7.6.2 Troubleshooting

When installing OFED-X.X.X, an error occurs saying that the installer failed to build the *ofa_kernel* RPM.

There is a chance that the kernel used by the server isn't supported by OFED. If the *Install All* option was chosen, try the *Customize* option in the OFED installation menu and select only the components that are needed. If this doesn't work, try installing a different version of OFED.

Loading the driver module failed (even after installing OFED).

Most likely the HCA used is not supported by OFED and the driver has not been installed correctly. Try visiting the specific HCA vendor's website to find the appropriate drivers for the card.

When trying to install OFED, an error occurs saying something similar to *<some_module> is required to build <some_other_module>.*

This means that there are some dependencies that are required by OFED that aren't installed on the server. The way to fix these problems is to install all of the required dependencies. If yum is available, use the following command to search for the installer needed:

```
# yum provides <dependency_name>
```

This will search for the specific name of the dependency's RPM that needs to be installed. Once the name of the RPM is in hand, type the following command to install it:

```
# yum install <dependency_rpm>
```

If yum is not installed on the server, each dependency will have to be manually downloaded and installed.

When trying to run *srp_daemon* an output occurs saying that a certain operation failed.

Make sure that the RamSan is physically connected to the network and that everything is powered on. Make sure that the right cable is being used and (if necessary) that OpenSM is running. To check if OpenSM is running, run the following command:

```
# /etc/init.d/opensmd status
```

Loading the *ib_srp* module fails.

Check that OFED is installed correctly and that the device drivers are also installed. If a custom OFED installation was performed,

make sure that *ibutils* and all packages related to *srp* were selected.

Chapter 8 – Solaris

Sun/Oracle Solaris will have some slight differences between x86 and SPARC support when partitioning the disks. The MPxIO multipathing setups, however, are identical.

8.1 Directly-Attached FC

When directly attached to the RamSan, the RamSan FC ports should be configured to Arbitrated Loop (AL) topology. If configured through a switch, the RamSan FC ports should be set to Point-to-Point to properly negotiate with the switch.

8.2 Sector Size

Currently, none of the Solaris versions support a 4 KB sector size. All LUNs presented to a Solaris host must be configured with a 512 byte sector size.

8.3 Alignment

Solaris SPARC will automatically align slices on LUNs when the LUN is using an SMI label for a LUN that is smaller than 2 TB. When breaching the 2 TB capacity, the OS must use an EFI label which will cause non-4 KB aligned slices.

EFI Disks will default to 34-sector offsets with the default partition table. To align the partition to a 4 KB boundary for optimal performance, change the first sector to 40 in the All Free Hog partition.

```
partition> mod
Select partitioning base:
   0. Current partition table (unnamed)
   1. All Free Hog
Choose base (enter number) [0]? 1
```

After writing the partition, change the *First Sector*, with <*enter*> for accepting defaults on all other options:

```
partition> 0
Enter partition id tag[usr]:
Enter partition permission flags[wm]:
Enter new starting Sector[34]: 40
Enter partition size[10066067388b, 10066067427e, 4915071mb,
4799gb, 4tb]:

Part      Tag    Flag  First Sector     Size     Last Sector
  0        usr    wm             40    4.69TB    10066067427
  1 unassigned    wm              0       0               0
```

```
  2 unassigned    wm               0       0               0
  3 unassigned    wm               0       0               0
  4 unassigned    wm               0       0               0
  5 unassigned    wm               0       0               0
  6 unassigned    wm               0       0               0
  8   reserved    wm    10066067422    8.00MB    10066083805
```

Once the first sector is changed, save the configuration and then continue using the newly created partition.

8.4 Multipathing

8.4.1 Solaris 8/9

MPxIO is the built-in multipathing mechanism for Solaris and requires a Sun-branded HBA. When using a non-Sun-branded HBA, then multipathing may be achieved by Veritas Storage Foundation.

To enable MPxIO support for RamSans, modify the */kernel/drv/scsi_vhci.conf* file with the following:

Note:

*There are **five (5) spaces** between* TMS *and* RamSan*.*

```
load-balance="round-robin";
mpxio-disable="no";

device-type-scsi-options-list =
"TMS     RamSan", "symmetric-option";
symmetric-option = 0x1000000;
```

8.4.2 Solaris 10

Just like Solaris 8/9, MPxIO is the built-in multipathing mechanism for Solaris and requires a Sun-branded HBA. When using a non-Sun-branded HBA, multipathing can be achieved by Veritas Storage Foundation.

To enable MPxIO support for RamSans, modify the */kernel/drv/scsi_vhci.conf* file with the following:

Note:

*There are **five (5) spaces** between* TMS *and* RamSan*.*

```
load-balance="round-robin";

device-type-scsi-options-list =
"TMS     RamSan", "symmetric-option";
symmetric-option = 0x1000000;
```

In addition, the */kernel/drv/fp.conf* will require:

```
mpxio-disable=”no”;
```

Chapter 9 – Veritas/Symantec Storage Foundation

Veritas/Symantec Storage Foundation also has *Just a Bunch of Disks* (JBOD) support for RamSans but requires the following command to acknowledge multiple RamSan LUNs. Run this command on the host.

```
# vxddladm addjbod vid="TMS" length=10
```

After entering this command, Storage Administrator will list the RamSan LUNs appropriately.

9.1 Sector Size

Versions of Storage Foundation through 5.1 do not support 4 KB sector sizes. Logical Units on the RamSan must be configured with a 512 byte sector size.

9.2 Alignment

When allocating disks through VEA, the default offset will usually be unaligned to a 4 KB boundary. To change this, use the *vxdisk* utility to create an offset and enforce track alignment on the disks before introducing them into a volume:

```
# vxdisk set track align=enable offset=128 <disk>
```

Note:

The above vxdisk *command only applies to Windows installations.*

9.3 Multipathing

The RamSan supports Active/Active multipathing through Veritas DMP.

9.4 I/O Recovery

To allow for successful failover in mirrored environments, lower the timeout threshold:

```
# vxdmpadm settune dmp_failed_io_threshold=60
```

Chapter 10 – VMWare ESX

ESX and ESXi are fairly limited as an OS, but the best practices must be carried through all levels of the application in order to achieve the optimal performance. The Guest OS best practices that can be found in other chapters of this guide are also required for optimal performance.

10.1 Directly-Attached FC

RamSan FCs should be configured to Arbitrated Loop (AL) topology when directly attached to the ESX hosts.

10.2 Sector Size

Currently, no version of ESX allows for 4 KB storage. LUNs configured for 4 KB sectors cannot be addressed as passthrough devices.

10.3 Alignment

VMWare will align its datastores to 64 KB appropriately, but guest VMs must still align their own presentation of the storage. Before continuing the installation of Linux or Windows Server 2003 Guest Operating Systems, partition the storage to the aligned accesses.

Reference the chapters on Linux and Windows regarding installing the OS on aligned partitions.

10.4 Multipathing

Round-robin multipathing is quickly configured through the *vSphere client->Configuration* tab. Under the *Devices* view, each disk can be selected to *Manage Paths…*. Under the *Manage Paths* dialog, change the *Path Selection* to *Round Robin (VMware)*.

Chapter 11 - HP-UX

HP-UX has a two-stage process to detect and initialize LUNs. The first process, # ioscan -C disk, will rescan for new disk devices. The second process, # ioinit –I, will create new Device Special Files (DSF) in the dev file system as /dev/dsk/cXtYdZ.

11.1 Directly-Attached FC

RamSan FCs should be configured to Arbitrated Loop (AL) topology when directly attached to the HP-UX hosts. For HP-UX 11.11, in order to maintain consistent device naming across server reboots, the FCs must be configured to a HARD Loop ID Assignment.

11.2 Sector Size

HP-UX does not support a 4 KB sector size. All Logical Units must be configured with a 512 byte sector size.

11.3 Alignment

HP-UX volumes will align to 4 KB boundaries. The VxFS file system should be set to a block size of 4096 or higher to keep alignment.

```
# mkfs -F vxfs -o bsize=4096 <disk>
```

11.4 Multipathing

HP-UX v1 and v2 include an Active/Passive Multipathing called PVLinks. The OS will report multiple disk devices, /dev/dsk/cYtXdZ, for a single Logical Unit with each device reflecting an FC path to the disk. To take advantage of PVLinks, add each disk/path into the same Volume Group under LVM.

HP-UX v3 has automatic multipathing with a round-robin policy. No additional steps are required to utilize the multipathing.

Chapter 12 - AIX

12.1 Directly-Attached FC

RamSan FCs should be configured to Arbitrated Loop (AL) topology when directly attached to the AIX hosts.

12.2 Sector Size

AIX does not support a 4 KB sector size. All Logical Units must be configured with a 512 byte sector size.

12.3 ACA Support

Enable ACA Support on LUNs connected to an AIX host. This can be done from the web interface's Logical Unit setup or from the command line interface. This enables Auto Contingent Allegiance (ACA) support for the Logical Unit. Some host systems, such as AIX, require this in order to run multiple concurrent commands. After this option is changed, all interface ports which have access to the logical unit must be reset.

12.4 Alignment

AIX volumes will align to 4 KB boundaries.

12.5 Multipathing

Texas Memory Systems, an IBM Company provides an ODM that will acknowledge the RamSan LUNs and create a new MPIO *hdisk*. Please see Appendix C for the ODM file.

If planning to enable the AIX MPIO for the RamSan LUN(s), follow these instructions prior to creating a volume.

12.5.1 Adding the ODM

After all paths are connected to the RamSan LUN(s), add the ODM definition that will define the RamSan device as one that supports multipathing.

Generate the ODM definition file from Appendix C by copying and pasting it into a standard text file.

To add the ODM definition to AIX, execute the following command:

```
odmadd <ODM Definition Filename>
```

Now execute the following to ensure it was added correctly:

```
lsdev -P | grep ramsan

Example:
# lsdev -P | grep ramsan
disk           ramsan         fcp        MPIO Other FC SCSI
Disk Drive
```

12.5.2 Resetting the HBA and Disk Configuration

After the ODM definition is added. The AIX system will need to rescan for the RamSan LUN(s) in order to recognize them as devices that support multipathing. It is often useful to start from scratch by reconfiguring the HBA and its attached disks.

Warning:

If there are other disks attached to any of these HBA devices the following commands will remove the configuration for those disks and the HBA itself. If attempting to save the current configuration, skip this step.

The first thing needed is the device name of the HBA(s) that the RamSan is connected to. To get the HBA device names, execute the following command:

```
lsdev -t efscsi

Example:
# lsdev -t efscsi
fscsi0 Available 1Z-08-01 FC SCSI I/O Controller Protocol
Device
```

For each HBA device name, execute the following command to remove the HBA and disk configuration associated with it:

```
rmdev -l <device name> -R

Example:
# rmdev -l fscsi0 -R
hdisk1 Defined
hdisk2 Defined
hdisk3 Defined
hdisk4 Defined
fscsi0 Defined
```

At this point there may be disks that were already defined, which should be removed before rescanning. If there are RamSan LUN(s)

already defined as "*Other FC SCSI Disk Drive*", remove the old definitions. For example:

```
# lsdev -C -c disk
hdisk0 Available 1S-08-00-8,0 16 Bit LVD SCSI Disk Drive
hdisk1 Defined 1Z-08-01     Other FC SCSI Disk Drive
hdisk2 Defined 1Z-08-01     Other FC SCSI Disk Drive
hdisk3 Defined 1Z-08-01     Other FC SCSI Disk Drive
hdisk4 Defined 1Z-08-01     Other FC SCSI Disk Drive
```

All four of these disks are paths to the same LUN. Delete their definitions using the following command:

```
rmdev -l <disk name> -d

Example:
# rmdev -l hdisk1 -d
hdisk1 deleted
```

12.5.3 Setting the Fast Fail Recovery Flag for the HBA

In order for the MPIO driver to fail over to an available path in a timely manner after a path failure, set the *fast_fail* recovery flag for the HBA device(s) that the RamSan is connected to.

For each HBA device name connected to the RamSan, execute the following command:

```
chdev -a fc_err_recov=fast_fail -l <device name>

Example:
# chdev -a fc_err_recov=fast_fail -l fscsi0
fscsi0 changed
```

12.5.4 Rescanning for the RamSan LUN(s)

Now that the system will recognize that the RamSan device supports multipathing, rescan for the LUN(s). To do so, execute the following command for each HBA device name:

```
cfgmgr -vl <device name>

Example:
# cfgmgr -vl fscsi0
----------------
attempting to configure device 'fscsi0'
Time: 0 LEDS: 0x569
invoking /usr/lib/methods/cfgefscsi -l fscsi0...
...
attempting to configure device 'hdisk1'
Time: 0 LEDS: 0x626
invoking /usr/lib/methods/cfgscsidisk -l hdisk1
...
```

```
Configuration time: 0 seconds
```

12.5.5 Checking the Configuration

To ensure that the configuration was successful, execute the following command to list all disks available to the system:

```
lsdev -C -c disk

Example:
# lsdev -C -c disk
hdisk0 Available 1S-08-00-8,0 16 Bit LVD SCSI Disk Drive
hdisk1 Available 1Z-08-01     MPIO Other FC SCSI Disk Drive
```

All RamSan LUN(s) should be shown as "*MPIO Other FC SCSI Disk Drive*".

If there are missing disks, or additional disks, or the RamSan LUN(s) do not show as "*MPIO Other FC SCSI Disk Drive*", check the connections and the RamSan configuration. The configuration for the HBA(s) attached to the RamSan may then need to be removed, and the rescan performed again (see *Section 11.4.2*).

Next, ensure that all the paths that are connected are visible. Execute the following command to list all paths:

```
lspath

Example:
# lspath
Enabled hdisk0 scsi0
Enabled hdisk1 fscsi0
Enabled hdisk1 fscsi0
```

If paths are missing, check the connections and RamSan configuration first. The configuration for the HBA(s) attached to the RamSan may then need to be removed, and the rescan performed again (see *Section 12.5.2*).

12.5.6 Active/Active

To setup a RamSan device to actively use all paths to it, set the *algorithm* attribute to *round_robin* before adding the *hdisk* to any volume group.

```
Example:
chdev -l hdisk1 -a algorithm=round_robin
```

12.5.7 Bosboot

To push the new disk configuration to the boot sector, the *bosboot* command may need to be run. This eliminates MPIO sub-devices from being displayed after server reboot.

After installing the ODM, save the system changes with the *bosboot* command:

```
bosboot
```

Chapter 13 – Oracle

13.1 Oracle ASM

Please refer to "SSD and Oracle ASM Best Practices for Using ASM with SSDs" for details on configuring the RamSan with ASM.

13.1.1 4 KB Sectors and Alignment

Oracle ASM supports the use of 4 KB sectors, limited to Linux when presented through ASMLib. Oracle ASM is aligned to 4 KB sector boundaries by default and does not require 4 KB alignment to achieve optimal performance.

If presenting a partitioned disk to Oracle ASM, it is crucial from a performance perspective for that partition to be aligned by following the recommend alignment steps for the OS.

All Oracle tablespaces, including User tablespaces, Undo, and Temp, perform I/Os at db_block_size multiples. As long as the database blocksize is 4 KB or higher for each tablespace, the I/O performance will be optimal.

The Oracle Redo and Control files cannot be forcibly aligned. For these particular files, RamSan systems contain a Log LUN feature to accelerate the latency of file operations. A separate diskgroup presented from Log LUN units from the RamSan should be created for the Oracle instances of these specific files.

Chapter 14 - Microsoft SQL Server

14.1 Disk and File System Setup

SQL Server performs best with an underlying sector size on the storage of 4 KB. With the sector size at 4 KB, it is still a best practice to format the NTFS file system at a 64 KB Allocation Unit (AU) size.

There is no noticeable performance advantage in separating database files, log files, and tempdb. However, it is recommended to separate these objects for easier management and performance monitoring.

14.2 Database Layout

It is a best practice to provision multiple database files (.ndf) for each database to address concurrent I/O. The Microsoft recommendation is one database file per CPU core.

This practice should also be applied to tempdb.

There is no advantage in performance for using multiple log files.

14.3 Moving Existing Database to 4 KB Sector Storage

14.3.1 Copy Database Wizard

If an existing database that was created on a storage device with 512 byte sectors is being migrated to the RamSan, the *Copy Database* wizard in SQL Server is the preferred practice. The database files are stamped with the sector size of their original disk, but this has not been used for any known supporting feature or application with SQL Server. The *Copy Database* wizard does include objects and references within the original SQL Server database.

14.3.2 Detach/Attach Database Files

For time-constraint deployments, the detach/attach workaround can be done with the following steps:

1. Detach the database.
2. Move only the .mdf and .ndf files to the new storage.
3. Delete the original .ldf log file.
4. Open the *Attach Database* dialog and select the .mdf datafile.
5. By default, the dialog will include the original .ldf file as *Not Found* -- remove this reference to the .ldf file so that the dialog will only include the .mdf and .ndf files.

6. Press OK.

There should be no errors and the SQL Server will create a new Log File for the instance. The new Log File is created at the SQL Server's default log file directory (SQL Server Properties -> Database Settings). To make the new Log File directory point to the RamSan file system, change the properties and restart the SQL Server instance to take effect.

14.4 MSCS + Veritas

Microsoft Cluster Services is commonly used for highly available SQL database deployments. TMS recommends mirroring RamSan systems when deployed in this environment. Veritas Storage Foundations is the supported cluster volume manager for MSCS and the RamSan. Please consult Chapter 9 for information on Veritas requirements.

This configuration can support advanced deployments including: preferred read mirroring between the RamSan and a slow traditional disks system, and network based replication. Please consult the *RamSan Veritas Storage Foundations and MCSC Setup Guide* for more details.

14.5 Database Mirroring

Database mirroring is a powerful feature within SQL server to create a high availability cluster or to add replication to a SQL database. Please consult the *RamSan Database Mirroring Best Practices* white paper for more details.

Chapter 15 – Sybase

15.1 4 KB Recommendations

The default page size for Sybase is 4 KB. The page size can be 2 KB, 4 KB, 8 KB, or 16 KB. Most write operations occur at 8 page collections called extents. For a 2 KB page size, an extent is 16 KB. Page size is universally set on the Master Database.

15.2 Disk Devices

Sybase works against RAW devices or file system devices.

```
/dev/mpath/<alias>

or

/work/Sybase/<filename>.dat
```

For raw devices, it is recommended to use the */dev/mpath/<alias>* for multipath targets, as the */dev/dm-X number* will vary and the */dev/mapper/<alias>* path will not function.

Devices can have a setting of *DSYNC*, *DirectIO*, or *CachedIO*.

- *DSYNC* should be used for devices when the page size is 2 KB.
- *DirectIO* should be used for devices when the page size is 4 KB or higher.
- *CachedIO* is optimized for disk drives, not for Flash storage such as the RamSan, and so is not recommended for the RamSan.

System Procedure *sp_logiosize* can be used to modify the log file write size to a 4 KB value for better performance and space efficiency.

15.3 Creating New Disk Devices

If not using Sybase Central (GUI) when creating new disk devices, here are the commands needed.

```
USE master
go
disk init name='rs1', physname='/dev/mpath/sybase1', vdevno=4,
size=2621440, cntrltype=0, directio = true
go
```

- */dev/mpath/sybase1* is a multipath disk soft-link.

- *vdevno* is the device enumeration in Sybase (which must be unique).
- *size* is in units of pages.
- *cntrltype* is rarely changed.
- *directio* enables DIRECTIO (which can also be DSYNC for 2 KB Pages).

15.4 Adding a Mirror Device

To add another RamSan as a mirror device, first enable Disk Mirroring and restart the database:

```
exec sp_configure 'disable disk mirroring', 0
go

<db restart>

USE master
go
disk mirror name='rs1', mirror='/dev/mpath/rs2', writes =
'noserial'
go
```

/dev/mpath/rs2 **cannot** be an already defined database device. Once mirrored, the database will read only from the primary copy of the mirror. This can also be used for preferred read mirroring.

Chapter 16 – Summary

The performance tweaks and best practices covered in this document will help get the highest performance from the RamSan Flash Systems. Some of the configuration settings depend on the I/O profile of the application using the RamSan Flash Systems. Questions regarding analysis and tuning for each application are welcome and encouraged: support@ramsan.com.

Appendix A – multipath.conf

Included below is the recommended 'multipath.conf' for Linux hosts referenced in Section 7.4. Copy and paste the following into a standard text file to use as the 'multipath.conf' file, or contact support@ramsan.com to receive a copy.

```
defaults {
        udev_dir                /dev
        polling_interval        30

# Linux 5.x: use the five parameters below (selector, getuid_callout,
#  prio_callout, rr_min_io, user_friendly_name).
        selector                "round-robin 0"                 # Linux 5.x
        getuid_callout "/sbin/scsi_id -g -u -s /block/%n"       # Linux 5.x
        prio_callout            "/bin/true"                     # Linux 5.x
        rr_min_io               1                               # Linux 5.x, RS-6xx/710/810
#       rr_min_io               4                               # Linux 5.x, RS-720/820
        user_friendly_name      yes                             # Linux 5.x

# Linux 6.x: use the five parameters below (path_selector, getuid_callout,
#  prio, rr_min_io_rq, user_friendly_names) instead of the five Linux 5.x
#  parameters above.
#       path_selector           "round-robin 0"                 # Linux 6.x
#       getuid_callout "/lib/udev/scsi_id --whitelisted --device=/dev/%n"   # Linux 6.x
#       prio                    const                           # Linux 6.x
#       rr_min_io_rq            1                               # Linux 6.x, RS-6xx/710/810
#       rr_min_io_rq            4                               # Linux 6.x, RS-720/820
#       user_friendly_names     yes                             # Linux 6.x

        path_grouping_policy    multibus
        path_checker            tur
        rr_weight               uniform
        failback                immediate
        no_path_retry           fail
}

blacklist {
        wwid                    "*"
}

blacklist_exceptions {
        # All RamSan WWIDs begins with this prefix.
        wwid                    "20020c24*"
}

multipaths {
        # Add a multipath section for each RamSan LUN as needed.
        multipath {
                # Change this example WWID to match the RamSan LUN.
                wwid            20020c24000093676
                # And give it an alias.
                alias           ramsan1
        }
}
```

Appendix B – Verifying Alignment

Alignment can be verified using the statistics on the RamSan Flash Systems. The RamSan statistics can be monitored through the *Web GUI->Statistics* tree-item. This can be done in real-time or to record the statistics for a later analysis.

The alignment-specific counters are logged at the port-level. When doing Active/Active multipathing, it is safe to review only one path as the other paths should retain the identical behavior. Be sure the path in review presents the LUNs to the application in question.

The counters specific for this analysis are as follows.

For RamSan firmware 5.3.2 or greater:

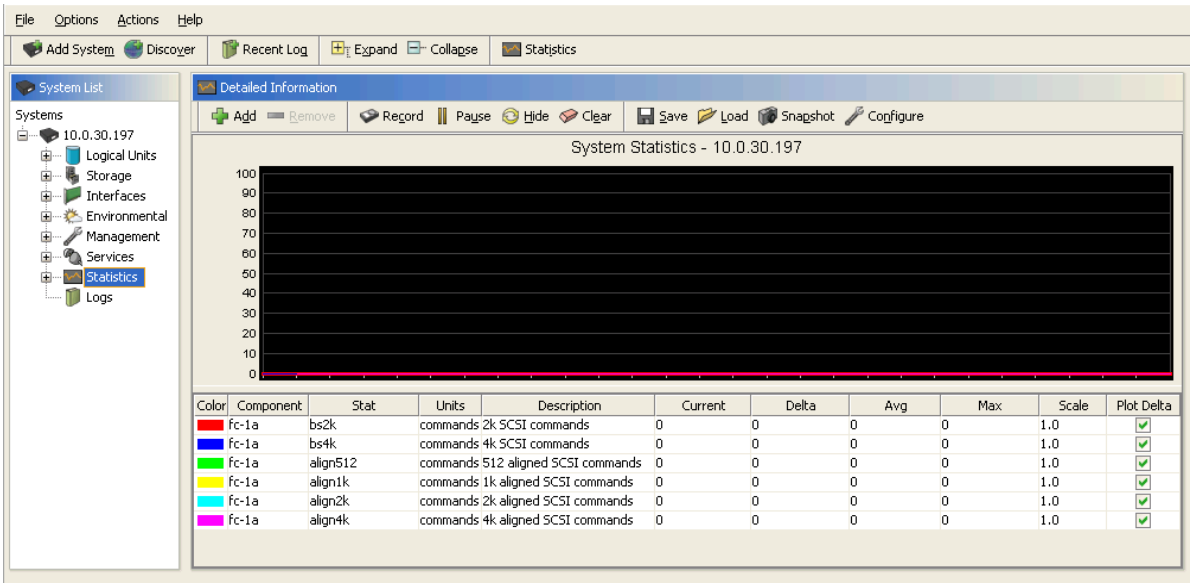
- *RMW Count*

For RamSan firmware prior to 5.3.2:

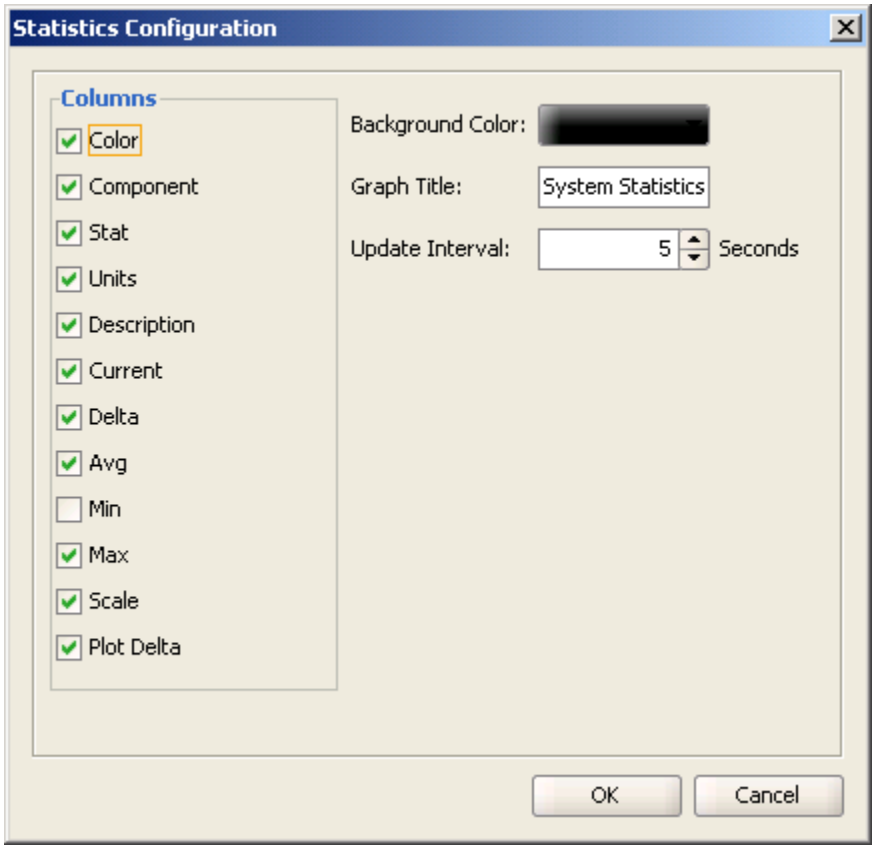
- *2k SCSI commands*
- *4k SCSI commands*
- *512 aligned SCSI commands*
- *1k aligned SCSI commands*
- *2k aligned SCSI commands*
- *4k aligned SCSI commands*

Read-Modify-Writes (RMW) are the direct result of unaligned writes, so for RamSan systems that support the *RMW Count* statistic it is the only one needed to verify alignment. The others can be used as well but are unnecessary for this exercise.

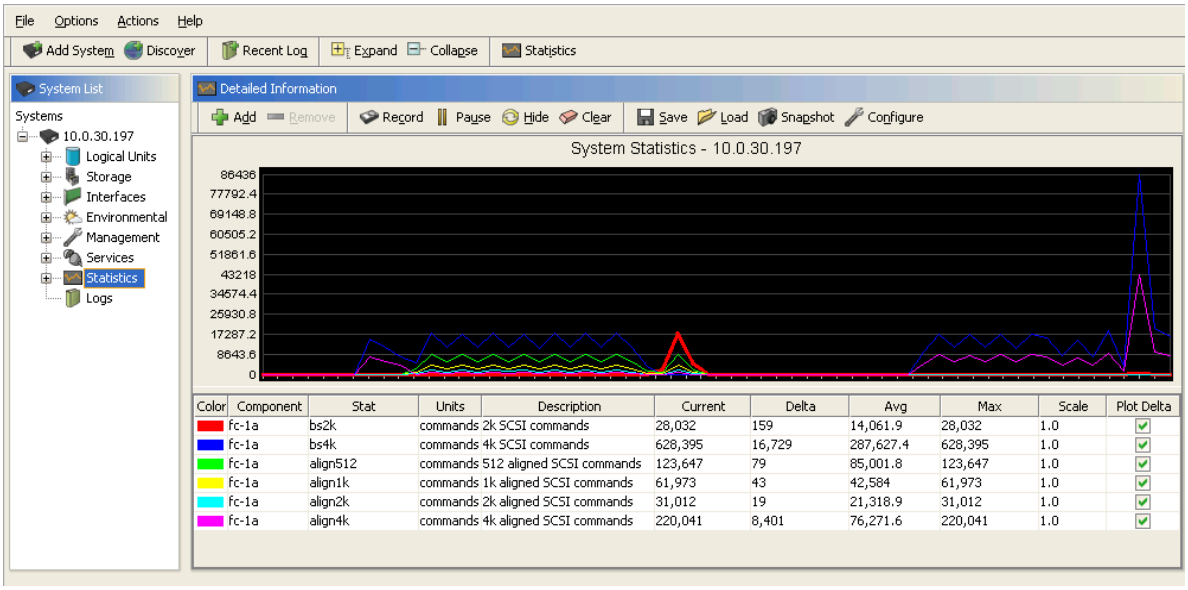
The *2k SCSI commands* and *4k SCSI commands* are counters for the size of the requests. The other four counters reference the starting address of the request. In order for an application to qualify as aligned, the size of the request must be a multiple of 4 KB and the starting address of the request must be divisible by 4 KB.



To also assist in the analysis, change the settings for the graph through the *Configure* toolbar button. Be sure to include the *Delta* column, *Plot Delta* column, and set the *Update Interval* to 5 seconds.



As the application runs, the counters will increase. Note that the counters are running counters and the *Current* column will only increase across samples. With the *Delta* column, the amount of commands received across samples is presented.



The *RMW Count*, the commands that are listed under the *2k SCSI commands,* and the commands that are below the 4k aligned SCSI commands (*512, 1k, 2k*) are all considered unaligned. As the objective of the application layout is to achieve alignment, and in-turn maximum application efficiency, the delta *RMW Count* should be minimal and the majority of the I/O should be accounted for by the 4k aligned counters.

Some unaligned I/Os may be unavoidable so do not expect a 100% aligned accesses.

These analyzed counters may also be extended to include the 8K, 16K, etc. variants. These variants are still aligned as they are at multiples of 4 KB.

If the application is presenting high delta *RMW Count* or does not present the majority of the I/O on the 4 KB aligned or higher counters, investigate partition alignment in reference to the OS chapter in the Integration Guide. Please contact support@ramsan.com for further assistance in the alignment and performance tuning.

Appendix C – AIX ODM File

Included below is the ODM file for AIX MPIO referenced in Section 12.5. Copy and paste the following into a standard text file to use as the ODM file, or contact support@ramsan.com to receive a copy.

```
PdDv:
        type = "ramsan"
        class = "disk"
        subclass = "fcp"
        prefix = "hdisk"
        devid = ""
        base = 1
        has_vpd = 1
        detectable = 1
        chgstatus = 0
        bus_ext = 0
        fru = 1
        led = 1574
        setno = 2
        msgno = 98
        catalog = "scdisk.cat"
        DvDr = "scsidisk"
        Define = "/usr/lib/methods/define"
        Configure = "/usr/lib/methods/cfgscsidisk"
        Change = "/usr/lib/methods/chgdisk"
        Unconfigure = "/usr/lib/methods/ucfgdevice"
        Undefine = "/usr/lib/methods/undefine"
        Start = ""
        Stop = ""
        inventory_only = 0
        uniquetype = "disk/fcp/ramsan"


PdPathAt:
       uniquetype = "disk/fcp/ramsan"
       attribute = "scsi_id"
       deflt = ""
       values = ""
       generic = "D"
       rep = "s"
       nls_index = 80

PdPathAt:
       uniquetype = "disk/fcp/ramsan"
       attribute = "node_name"
       deflt = ""
       values = ""
       generic = "D"
       rep = "s"
       nls_index = 94

PdAt:
        uniquetype = "PCM/friend/fcpother"
        attribute  = "dvc_support"
        deflt      = ""
        values     = "disk/fcp/ramsan"
        width      = ""
```

```
        type        = "R"
        generic     = ""
        rep         = "sl"
        nls_index   = 2

PdAt:
       uniquetype = "disk/fcp/ramsan"
       attribute = "scsi_id"
       deflt = ""
       values = ""
       width = ""
       type = "R"
       generic = "D"
       rep = "s"
       nls_index = 80

PdAt:
       uniquetype = "disk/fcp/ramsan"
       attribute = "lun_id"
       deflt = ""
       values = ""
       width = ""
       type = "R"
       generic = "D"
       rep = "s"
       nls_index = 81

PdAt:
       uniquetype = "disk/fcp/ramsan"
       attribute = "location"
       deflt = ""
       values = ""
       width = ""
       type = "R"
       generic = "DU"
       rep = "s"
       nls_index = 82

PdAt:
       uniquetype = "disk/fcp/ramsan"
       attribute = "ww_name"
       deflt = ""
       values = ""
       width = ""
       type = "R"
       generic = "D"
       rep = "s"
       nls_index = 83

PdAt:
       uniquetype = "disk/fcp/ramsan"
       attribute = "node_name"
       deflt = ""
       values = ""
       width = ""
       type = "R"
       generic = "D"
       rep = "s"
       nls_index = 94

PdAt:
       uniquetype = "disk/fcp/ramsan"
       attribute = "pv"
```

```
        deflt = "yes"
        values = "no,yes,clear"
        width = ""
        type = "R"
        generic = "U"
        rep = "sl"
        nls_index = 0

PdAt:
        uniquetype = "disk/fcp/ramsan"
        attribute = "pvid"
        deflt = "none"
        values = ""
        width = ""
        type = "R"
        generic = "D"
        rep = "s"
        nls_index = 2

PdAt:
        uniquetype = "disk/fcp/ramsan"
        attribute = "mode_data"
        deflt =
"0x00000008000000000000020001012402 0a000000000000000000000070a050000000000000000
00080c0000ffff0000ffffffff00040a06000100000000"
        values = ""
        width = ""
        type = "R"
        generic = ""
        rep = "s"
        nls_index = 0

PdAt:
        uniquetype = "disk/fcp/ramsan"
        attribute = "mode_default"
        deflt = "0x00000000010700ffff000000ff0202ffff070a00ffff0000000000ffff"
        values = ""
        width = ""
        type = "R"
        generic = ""
        rep = "s"
        nls_index = 0

PdAt:
        uniquetype = "disk/fcp/ramsan"
        attribute = "recovery_limit"
        deflt = "10"
        values = "10"
        width = ""
        type = "R"
        generic = ""
        rep = "n"
        nls_index = 0

PdAt:
        uniquetype = "disk/fcp/ramsan"
        attribute = "safe_relocate"
        deflt = "0"
        values = "0"
        width = ""
        type = "R"
        generic = ""
        rep = "n"
```

```
        nls_index = 0

PdAt:
        uniquetype = "disk/fcp/ramsan"
        attribute = "reset_delay"
        deflt = "2"
        values = "2"
        width = ""
        type = "R"
        generic = ""
        rep = "n"
        nls_index = 0

PdAt:
        uniquetype = "disk/fcp/ramsan"
        attribute = "queue_depth"
        deflt = "256"
        values = "1-256,1"
        width = ""
        type = "R"
        generic = "UD"
        rep = "nr"
        nls_index = 30

PdAt:
        uniquetype = "disk/fcp/ramsan"
        attribute = "q_type"
        deflt = "simple"
        values = "none,ordered,simple"
        width = ""
        type = "R"
        generic = "UD"
        rep = "sl"
        nls_index = 31

PdAt:
        uniquetype = "disk/fcp/ramsan"
        attribute = "q_err"
        deflt = "yes"
        values = "no,yes"
        width = ""
        type = "R"
        generic = "UD"
        rep = "sl"
        nls_index = 32

PdAt:
        uniquetype = "disk/fcp/ramsan"
        attribute = "clr_q"
        deflt = "no"
        values = "no,yes"
        width = ""
        type = "R"
        generic = "UD"
        rep = "sl"
        nls_index = 33

PdAt:
        uniquetype = "disk/fcp/ramsan"
        attribute = "ses_attach"
        deflt = "no"
        values = "no,yes"
        width = ""
```

```
        type = "R"
        generic = ""
        rep = "sl"
        nls_index = 0

PdAt:
        uniquetype = "disk/fcp/ramsan"
        attribute = "rw_timeout"
        deflt = "30"
        values = "30-1000,1"
        width = ""
        type = "R"
        generic = "UD"
        rep = "nr"
        nls_index = 34

PdAt:
        uniquetype = "disk/fcp/ramsan"
        attribute = "start_timeout"
        deflt = "60"
        values = "30-1000,1"
        width = ""
        type = "R"
        generic = "UD"
        rep = "nr"
        nls_index = 35

PdAt:
        uniquetype = "disk/fcp/ramsan"
        attribute = "max_transfer"
        deflt = "0x40000"
        values =
"0x20000,0x40000,0x80000,0x100000,0x200000,0x400000,0x800000,0x1000000"
        width = ""
        type = "R"
        generic = "DU"
        rep = "nl"
        nls_index = 88

PdAt:
        uniquetype = "disk/fcp/ramsan"
        attribute = "reassign_to"
        deflt = "30"
        values = "30-1000,1"
        width = ""
        type = "R"
        generic = "UD"
        rep = "nr"
        nls_index = 36

PdAt:
        uniquetype = "disk/fcp/ramsan"
        attribute = "pm_dev_itime"
        deflt = "28800"
        values = "0-28800,1"
        width = ""
        type = "R"
        generic = ""
        rep = "nr"
        nls_index = 0

PdAt:
        uniquetype = "disk/fcp/ramsan"
```

```
        attribute = "pm_dev_stime"
        deflt = "28800"
        values = "0-28800,1"
        width = ""
        type = "R"
        generic = ""
        rep = "nr"
        nls_index = 0

PdAt:
        uniquetype = "disk/fcp/ramsan"
        attribute = "pm_devid"
        deflt = "0x00100000"
        values = "0x00100000,
0x00180000,0x00181000,0x00182000,0x00183000,0x00184000,0x00185000,
0x00186000,0x00187000,0x00188000,0x00189000,0x0018a000,0x0018b000,0x0018c000,0x
0018d000,0x0018e000,0x0018f000"
        width = ""
        type = "R"
        generic = ""
        rep = "nr"
        nls_index = 0

PdAt:
        uniquetype = "disk/fcp/ramsan"
        attribute = "cfgmgr_psafe"
        deflt = ""
        values = ""
        width = ""
        type = "R"
        generic = ""
        rep = ""
        nls_index = 0

PdAt:
        uniquetype = "disk/fcp/ramsan"
        attribute = "reserve_policy"
        deflt = "no_reserve"
        values = "no_reserve,single_path"
        width = ""
        type = "R"
        generic = "DU"
        rep = "sl"
        nls_index = 96

PdAt:
        uniquetype = "disk/fcp/ramsan"
        attribute = "PCM"
        deflt = "PCM/friend/fcpother"
        values = "PCM/friend/fcpother"
        width = ""
        type = "_FD"
        generic = "DU"
        rep = "sl"
        nls_index = 97

PdAt:   uniquetype = "disk/fcp/ramsan"
        attribute = "hcheck_mode"
        deflt = "enabled"
        values = "enabled,failed,nonactive"
        width = ""
        type = "R"
        generic = "DU"
```

```
        rep = "sl"
        nls_index = 5

PdAt:   uniquetype = "disk/fcp/ramsan"
        attribute = "hcheck_interval"
        deflt = "20"
        values = "0-3600,1"
        width = ""
        type = "R"
        generic = "DU"
        rep = "nr"
        nls_index = 7

PdAt:
       uniquetype = "disk/fcp/ramsan"
       attribute = "model_map"
       deflt = "080ETMS     RamSan"
       values = "240A"
       width = ""
       type = "R"
       generic = "D"
       rep = "s"
       nls_index = 0

PdAt:
        uniquetype = "disk/fcp/ramsan"
        attribute = "model_map"
        deflt = "080CTMS     FC65"
        values = "240A"
        width = ""
        type = "R"
        generic = "D"
        rep = "s"
        nls_index = 0

PdAt:
       uniquetype = "disk/fcp/ramsan"
       attribute = "unique_id"
       deflt = ""
       values = "240A"
       width = ""
       type = "R"
       generic = "D"
       rep = "nl"
       nls_index = 79
```