## Document revision

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>04/2/2012</td>
<td>B</td>
<td>Updated Veritas Section</td>
</tr>
</tbody>
</table>

THIS WHITE PAPER IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND.

© 2012 Dell Inc. All rights reserved. Reproduction of this material in any manner whatsoever without the express written permission of Dell Inc. is strictly forbidden. For more information, contact Dell.

*Dell*, the *DELL* logo, the *DELL* badge, and Compellent are trademarks of Dell Inc. Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Dell Inc. disclaims any proprietary interest in trademarks and trade names other than its own.
Contents

Document revision ........................................................................................................... 2
General syntax .................................................................................................................. 6
Conventions ...................................................................................................................... 6
Preface ............................................................................................................................... 7
Audience ........................................................................................................................... 7
Purpose ............................................................................................................................. 7
Customer support ........................................................................................................... 7
Introduction ....................................................................................................................... 8
Useful Tools ....................................................................................................................... 8
  format ............................................................................................................................. 8
  fcinfo ............................................................................................................................. 8
  devfsadm ....................................................................................................................... 9
  cfgadm .......................................................................................................................... 9
  luxadm ......................................................................................................................... 10
  mpathadm ..................................................................................................................... 10
Fibre Channel .................................................................................................................... 10
  Fibre Channel Drivers ................................................................................................. 11
Managing Volumes ............................................................................................................ 12
  Scanning for New Volumes ......................................................................................... 13
  Correlating Storage Center Serial Number with OS Device ...................................... 16
Removing Volumes .......................................................................................................... 18
Partitions and File Systems ............................................................................................... 20
  Partitions/Slices .......................................................................................................... 20
  Disk Labels .................................................................................................................. 20
  SMI Labels ................................................................................................................... 20
  EFI Labels .................................................................................................................... 20
  UFS File System .......................................................................................................... 20
  Expanding a UFS File System ..................................................................................... 22
  UFS and Data Instant Replay ...................................................................................... 27
  UFS Performance Tuning ............................................................................................. 30
  ZFS File System .......................................................................................................... 33
  Mapping a Volume for ZFS ......................................................................................... 33
  Expanding a ZFS File System ...................................................................................... 35
  ZFS and Data Instant Replay ....................................................................................... 38
Dell Compellent Storage Center Solaris Best Practices

ZFS Performance Tuning ................................................................. 40
Multipath with Solaris 10 .................................................................. 41
Configuring and Activating Multipath (MPxIO) on Solaris 10 with Storage Center ................................................................. 41
Matching the MPxIO Device Name with /dev/rdsk Device Entry .......... 43
Dell Compellent Storage Center Controller Failover Timing .................. 43
Software iSCSI .............................................................................. 44
Mapping an iSCSI Volume to a Solaris 10 Host .................................. 45
Remove an iSCSI Volume from a Solaris 10 Host ................................. 52
Remove an iSCSI Discovery Address from a Solaris 10 Host .................. 54
Multiple iSCSI Paths (MPxIO) vs. Single Path iSCSI Performance Comparison ............................................................... 55
Scripting/Automation with CompCU ................................................. 55
Verifying Java and Initially Configuring and Testing CompCU Functions ............................................................... 56
Using CompCU to Automate Common Tasks ...................................... 58
Appendix ....................................................................................... 64
Boot from SAN ............................................................................. 64
Open Boot Prom (OBP) ................................................................. 65
Veritas ........................................................................................... 68

Tables
Table 1. Document syntax .................................................................. 6

Figures
Figure 1. Solaris 10 Format Output .................................................... 16
Figure 2. Storage Center Volume Serial Number .................................. 17
Figure 3. Volume Size Before Expansion ........................................... 23
Figure 4. Volume Size After Expansion ............................................... 23
Figure 5. Dell Compellent Volume Replay Tab ...................................... 28
Figure 6. Create View Volume ........................................................... 29
Figure 7. Map View Volume Summary Screen ...................................... 29
Figure 8. Example Storage Center iSCSI Ports ...................................... 45
Figure 9. Example Server Object Creation - iSCSI IQN Port Information ............................................................... 48
Figure 10. Example Server Object Creation - iSCSI Object Name and OS Type ............................................................... 49
Figure 11. Example Server Object Creation - iSCSI Object Attributes ............................................................... 50
Figure 12. Example Mapping to iSCSI Server Object ............................... 50
Figure 13. CompCU Generated Single Volume Example .......................... 59
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>Replay Listed Created from CompCU</td>
<td>60</td>
</tr>
<tr>
<td>15</td>
<td>CompCU Generated Multiple Volumes Example</td>
<td>63</td>
</tr>
<tr>
<td>16</td>
<td>Veritas - Output of vxddladm Command</td>
<td>69</td>
</tr>
<tr>
<td>17</td>
<td>Veritas - Install a Product Screenshot</td>
<td>70</td>
</tr>
<tr>
<td>18</td>
<td>Veritas - Install Program Screenshot</td>
<td>71</td>
</tr>
<tr>
<td>19</td>
<td>Veritas - Install Packages Screenshot</td>
<td>71</td>
</tr>
<tr>
<td>20</td>
<td>Veritas - Installation Progress Screenshot</td>
<td>72</td>
</tr>
<tr>
<td>21</td>
<td>Veritas - Installation Completion Screenshot</td>
<td>72</td>
</tr>
<tr>
<td>22</td>
<td>Veritas - Services Startup Screenshot</td>
<td>73</td>
</tr>
<tr>
<td>23</td>
<td>Veritas - Host Reported Utilization - Pre File Deletes</td>
<td>79</td>
</tr>
<tr>
<td>24</td>
<td>Veritas - Storage Center Reported Utilization - Pre thinrclm</td>
<td>80</td>
</tr>
<tr>
<td>25</td>
<td>Veritas - Host Reported Utilization - Post File Deletes</td>
<td>80</td>
</tr>
<tr>
<td>26</td>
<td>Veritas - Storage Center Utilization - Pre thinrclm, Post File Deletes</td>
<td>81</td>
</tr>
<tr>
<td>27</td>
<td>Veritas - fsadm Execution Example</td>
<td>81</td>
</tr>
<tr>
<td>28</td>
<td>Veritas - Storage Center Statistics for Volume After fsadm</td>
<td>82</td>
</tr>
<tr>
<td>29</td>
<td>Veritas - Storage Center Volume Size after Expanded</td>
<td>83</td>
</tr>
</tbody>
</table>
**General syntax**

**Table 1. Document syntax**

<table>
<thead>
<tr>
<th>Item</th>
<th>Convention</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu items, dialog box titles, field names, keys</td>
<td><strong>Bold</strong></td>
</tr>
<tr>
<td>Mouse click required</td>
<td><strong>Click:</strong></td>
</tr>
<tr>
<td>User Input</td>
<td><strong>Monospace Font</strong></td>
</tr>
<tr>
<td>User typing required</td>
<td><strong>Type:</strong></td>
</tr>
<tr>
<td>Website addresses</td>
<td><a href="http://www.compellent.com">http://www.compellent.com</a></td>
</tr>
<tr>
<td>Email addresses</td>
<td><a href="mailto:info@compellent.com">info@compellent.com</a></td>
</tr>
</tbody>
</table>

**Conventions**

- **Note**
  
  Notes are used to convey special information or instructions.

- **Timesaver**
  
  Timesavers are tips specifically designed to save time or reduce the number of steps.

- **Caution**
  
  Caution indicates the potential for risk including system or data damage.

- **Warning**
  
  Warning indicates that failure to follow directions could result in bodily harm.
Preface

Audience
The audience for this document is System Administrators who are responsible for the management of Solaris systems that utilize the Dell Compellent Storage Center SAN. Although there are references and comments to Solaris 8 and 9, the focus of this white paper is on the Solaris 10 release as that is the most recent release of Solaris at the time of publication. Also, this document will focus exclusively on the SPARC platform for Solaris. Environments utilizing the Solaris x86 variant can also benefit from the information in this guide, this guide is written with the SPARC platform as the base reference architecture.

All examples depicted in this paper were run on a Solaris 10 Update 9 SPARC based system. Two systems were used at varying points when multiple systems were needed for demonstration purposes (such as to demonstrate Data Instant Replay). The primary system used was an Oracle Sun T2000, with a secondary system that was an Oracle Sunfire v120. The same Solaris 10 Update 10, with identical 10_Recommended patch cluster applied from October 2011.

Purpose
This document is intended to provide an overview of Solaris Best Practices when using the Dell Compellent Storage Center. For additional Solaris Best Practices and Technical Tips, download these from http://knowledgecenter.compellent.com.

This document is intended for administrators with at minimum a basic understanding of Solaris systems, specifically general tasks around managing disk partitions and file systems.

It is important to note that as is common in Solaris, there are many ways to do what is covered in this document. This guide does not contain every possible way, and the way covered might not be the best for all situations. This documentation is brief and intended as a starting point of reference for end users. Users are encouraged to consult more detailed documentation available from Oracle.

Also note that this guide will focus almost exclusively on the command line. There are graphical tools to achieve many of these tasks. This guide simply focuses on the command line because it is the most universal.

Customer support
Dell Compellent provides live support 1-866-EZSTORE (866.397.8673), 24 hours a day, 7 days a week, 365 days a year. For additional support, email Dell Compellent at support@compellent.com. Dell Compellent responds to emails during normal business hours.
Introduction

The goal of this paper is to provide guidance to Solaris system administrators who will be utilizing storage presented from the Dell Compellent Storage Center SAN. It is also intended to be useful for storage administrators who manage the Dell Compellent SAN in environments that have Solaris-based hosts connecting to the SAN.

Useful tools

The Dell Compellent Storage Center SAN does not require the installation of any additional software beyond what is provided with the base Operating System. There are several OS utilities that will be leveraged in order to manage LUNs presented to a Solaris host. These commands/utilities are discussed below in some detail. Please refer to the online manpage or the Oracle website for further details.

A great place to start would be http://www.oracle.com/technetwork/documentation/solaris-10-192992.html

format

This is the disk partitioning and maintenance utility for Solaris on SPARC. This tool is used to label the newly provisioned LUN and create partitions. Note that for ZFS-based LUNs it is recommended to not create partitions, but rather use the entire disk when creating the zfs pool and file system.

{root@vito} {/} # format
Searching for disks...done

AVAILABLE DISK SELECTIONS:

0. c4t6000D310000069000000000000000F10d0 <COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 126>
/scsi_vhci/ssd@g6000d310000069000000000000000f10
Specify disk (enter its number):

fcinfo

The fcinfo command is used to gather information about Fibre Channel HBA devices installed in the Solaris system. This command is useful when needing to cross-reference the WWN reported on the host with the WWN reported for the HBA in Storage Center.

{root@fiero} {/} # fcinfo hba-port
HBA Port WWN: 210000e08b17c800
OS Device Name: /dev/cfg/c0
Manufacturer: QLogic Corp.
Model: QLA2342
Firmware Version: 03.03.28
FCode/BIOS Version: fc code: 1.17;
Serial Number: not available
Driver Name: qlc
Driver Version: 3.00p
Type: N-port
State: online
Supported Speeds: 1Gb 2Gb
Current Speed: 2Gb
Node WWN: 200000e08b17c800
HBA Port WWN: 210100e08b37c800
OS Device Name: /dev/cfg/c1
Manufacturer: QLogic Corp.
Model: QLA2342
Firmware Version: 03.03.28
FCode/BIOS Version: fcode: 1.17;
Serial Number: not available
Driver Name: qlc
Driver Version: 3.00p
Type: N-port
State: online
Supported Speeds: 1Gb 2Gb
Current Speed: 2Gb
Node WWN: 200100e08b37c800

devfsadm
The devfsadm command is used to manage the device tree listed in the /dev structure. This is used as part of the LUN management, specifically when it is needed to add and/or remove /dev links to LUNs that have been removed from the system.

{root@vito} {/} # devfsadm -Cv

cfgadm
The cfgadm command is used to manage the configuration of devices in the Solaris system. In particular, when a volume is mapped or unmapped from Storage Center, cfgadm is used to scan the controller ports of the HBA to (un)discover volumes. The example below illustrates a typical command to scan the Fibre Channel HBA ports of a system to discover a newly mapped volume.

To discover a mapped volume on device path /dev/cfg/c0, for example, simply run the command below:

{root@fiero} {/} # cfgadm -c configure c0

After discovery a mapped volume, the system log will report output similar to below:

{root@fiero} {/} # tail -f /var/adm/messages
Nov 3 14:28:55 fiero scsi: [ID 107833 kern.warning] WARNING:
/scsi_vhci/ssd@g6000d31000006900000000000000000f0f (ssd1):
Error Level: Retryable
Nov 3 14:28:55 fiero scsi: [ID 107833 kern.notice] Requested Block: 4208136 Error Block: 4208136
Nov 3 14:28:55 fiero scsi: [ID 107833 kern.notice] Vendor: COMPELNT
Serial Number:
Nov 3 14:28:55 fiero scsi: [ID 107833 kern.notice] Sense Key: Unit Attention
Nov 3 14:28:55 fiero scsi: [ID 107833 kern.notice] ASC: 0x3f (reported LUNs data has changed), ASCQ: 0xe, FRU: 0x0
Nov 3 14:28:55 fiero scsi: [ID 799468 kern.info] ssd3 at scsi_vhci0: name g6000d3100000690000000000000000004e, bus address g6000d31000006900000000000000000000f4e
Nov 3 14:28:55 fiero genunix: [ID 936769 kern.info] ssd3 is /scsi_vhci/ssd@g6000d31000006900000000000000000004e
Nov 3 14:28:55 fiero scsi: [ID 107833 kern.warning] WARNING:
/scsi_vhci/ssd@g6000d31000006900000000000000000000f4e (ssd3):
Corrupt label; wrong magic number
Nov 3 14:28:55 fiero genunix: [ID 408114 kern.info] /scsi_vhci/ssd@g6000d310000069000000000000000000000f4e (ssd3) online
The luxadm command is used to manage specific aspects of a LUN, such as to bring a LUN offline or to gather detailed information about a given LUN. The example below illustrates the usage of luxadm to bring a LUN offline in preparation to have it removed from the Solaris system, such as in the case of a View volume of a Replay having been mapped to recover some data and is no longer needed.

```
(root@fiero) {/} # luxadm -e offline
/dev/rdsk/c4t6000D31000006900000000000000F4Ed0s0
```

The mpathadm command is used to manage and display information regarding the multipath state of the attached LUNs. This command is used in conjunction with the Sun Solaris SAN Foundation Suite (SFS), including MPxIO.

```
(root@vito) {/} # mpathadm list lu
/dev/rdsk/c4t6000D31000069000000000000000F10d0s2
Total Path Count: 2
Operational Path Count: 2
```

**Fibre Channel**

The Dell Compellent Storage Center SAN supports both Fibre Channel Protocol (FCP) and iSCSI attached LUNs. Generally speaking, most Solaris SPARC environments have traditionally used FCP to connect Solaris servers to the SAN. This section will focus on concepts pertaining directly to the FCP model.

The Dell Compellent SAN utilizes the native tools provided by the OS vendor to provide the interconnectivity between the host and the SAN. It is recommended to configure the environment with multipath whenever possible. For details on this, please refer to the Multipath section below.

There are two Fibre Channel HBA brands that are supported on the Solaris SPARC platform. These are models from QLogic and Emulex. Both of these types of HBAs can come with either the native HBA vendor firmware or the Sun-branded firmware. It is recommended for most environments to use the Sun-branded firmware if installed on the HBA.

The QLogic and Sun-QLogic adapters look the same physically. However, Sun-QLogic adapters include Sun firmware, the Sun label, and a part number beginning with SG instead of QLA. The same is true with the Emulex and Sun-Emulex adapters.

The Solaris-supplied SFS software can control both Vendor-supplied and Solaris-supplied firmware on the HBA. This allows for standardization of drivers across the Solaris environment.

The recommended Fibre Channel driver for use with Dell Compellent Storage Center is the Sun SFS with MPxIO.
Since Oracle develops the operating system for its servers, Oracle is in the ideal position to develop a better I/O connection, or stack, into its servers. In essence, Oracle is providing a clean, well standardized, open interface point at the SAN port on Sun servers with single vendor accountability, reducing complexity and simplifying support for customers who choose to use storage from Dell Compellent.

Oracle provides a standards-based I/O framework and device driver stack called the Solaris SAN Foundation Suite (SFS), commonly known by its code name Leadville. It includes the Leadville I/O stack, MPxIO multipathing, (SNIA) libraries, and utilities. The Leadville stack is fully integrated into the Solaris 10 Update 2, a.k.a. Version 6/06 and later releases. Some further information on SFS can be found here:

http://wikis.sun.com/display/SFSRelNotes/Sun+StorEdge+SFS+4.4.15+Release+Notes

**SUN Microsystems SFS-FCA (QLC/EMLXS) Driver**
http://developers.sun.com/solaris/articles/fc_drivers_sfs.html

In addition, QLogic provides an open driver called QLA, which operates in the Solaris environment as a third party add on Fiber Channel driver. The QLA driver for Solaris 8, 9 and 10 are available on the QLogic website. Please contact Qlogic for specific details on QLA support and operation.

**Fibre Channel drivers**

There are two HBA vendor models supported with the Solaris 10 Operating System and the Dell Compellent Storage Center SAN. These are from QLogic and Emulex respectively. The section below details some of the specific information and recommendations surrounding these HBAs. The information below is not intended to be a replacement for vendor-specific or Oracle-specific documentation.

**SFS-FCA-QLC**

The QLC driver is installed by default starting with Solaris 10 Update 2. For Solaris 8 and 9, this is installed as an add-on via multiple OS patches, or by using the “install_it” script from the StorEdge SAN software.

The QLC driver discovers direct-attach and loop targets, but not fabric attached targets, by default. On Solaris 10 systems, to connect new fabric attached targets, use the `cfgadm` utility in order to discover the newly mapped volume(s). Refer to the Useful Tools section above for additional information.

On Solaris 8 and 9 servers the controller number and port information for each Fiber Channel card is listed in the Solaris `/var/adm/messages` file or via `# dmesg`.

Dell Compellent recommends using the OS-supplied QLC driver with QLogic Fibre Channel HBAs and Storage Center.

**Dell Compellent Specific Settings for the QLC Driver**

In a single path configuration, Dell Compellent recommends the following settings for the Solaris 10 QLC driver. It is important to allow enough time for a volume to failover from one Storage Center controller to the other; sufficient time should be allowed for
this failover process to complete. It can take up to 60 seconds for this to occur, depending on the number of volumes that need to be transferred.

Therefore, it is recommended to adjust the Port Down Retry Count and Login Retry Count parameters in the `/kernel/drv/qlc.conf` file. After making the following changes in the file, the system should be rebooted to activate the changes:

- `port-down-retry-count=60;`
- `login-retry-count=60;`

In multipath configurations, these values can typically be set to lower values, please refer to the Multipathing section for details on recommended settings when using the MPxIO multipath driver with Dell Compellent Storage Center.

**QLoCic QLA**
The QLA driver is a Qlogic supplied Legacy driver in the Solaris package file format. To check if the QLA driver is installed on the Sun server, run `#pkginfo | grep QLA`.

Please contact Qlogic support for the latest drivers, configuration and features for the Qlogic third party QLA driver when connecting your Solaris server to the Storage Center. The SanSurfer utility from Qlogic can be used to update firmware on Qlogic fiber channel HBA’s.

**SFS-FCA-emlxs**
The EMLXS driver is installed by default starting with Solaris 10 Update 2. This is the newer driver for use with Emulex based Fibre Channel HBAs and is planned to be continually updated by Emulex. Dell Compellent Best Practices calls for the use of the Solaris supplied SFS-FCA-emlxs driver. In addition, the ability to boot from SAN is activated automatically with the SFS-emlxs driver takes control of the fiber channel HBA.

**Dell Compellent Specific Settings for the EMLXS Driver**
There are currently no custom recommendations for this driver.

**Emulex SD LPFC**
The Emulex SD LPFC driver is a legacy driver with no planned extensions beyond current capabilities. If this driver is intended for use, please refer to the Emulex documentation for proper configuration and usage limitations. Dell Compellent does not recommend using this driver.

**Managing Volumes**
Understanding how volumes are managed in the Solaris system requires a basic understanding of the Solaris device structure. The `/devices` directory contains the physical device file for every hardware device attached to the Solaris system. In addition, there is the `/dev` directory, which contains a `dsk` and `rdsk` subdirectories. These sub-directories have symbolic links that reference the block and raw device paths listed in the `/devices` tree. Fibre Channel attached devices will be found under the `/devices/scsi_vhci` device tree and will utilize the ssd device driver.

Any volume mapped to the Solaris host will be present under this path as a LUN and be presented in the format: `ssd@port,target:partition`
To illustrate this, in the example below, the root OS file system is located on the SAN in a boot-from-SAN configuration, configured as a multipathed device. The device is presented in the /devices/scsi_vhci path as follows:

```
# What is the root file system device
[root@fiero] {/} # df /
File system size used avail capacity Mounted on
/dev/dsk/c4t6000d310000069000000000000000F0Fd0s0 21G 12G 8.8G 58% /
```

```
# List the root device in /devices tree:
[root@fiero] {/} # ll /devices/scsi_vhci/
total 2
brw-r----- 1 root sys 118, 8 Nov 3 13:38
ssd@g6000d310000069000000000000000F0Fd0s0
```

```
# List the reference to the /devices entry in /dev/dsk and /dev/rdsk:
[root@fiero] {/} # ll /dev/dsk/c4*
lrwxrwxrwx 1 root root 63 Aug 15 15:17 
/dev/dsk/c4t6000d310000069000000000000000F0Fd0s0 -> 
.../devices/scsi_vhci/ssd@g6000d310000069000000000000000F0Fd0s0:
```

```
[root@fiero] {/} # ll /dev/rdsk/c4*
lrwxrwxrwx 1 root root 67 Aug 15 15:17 
/dev/rdsk/c4t6000d310000069000000000000000F0Fd0s0 -> 
.../devices/scsi_vhci/ssd@g6000d310000069000000000000000F0Fd0s0,raw
```

In addition, there are two volume label types for Solaris 10 SPARC platforms. The traditional SMI label, which consists of eight (8) slices, numbered 0 to 7, and is limited to 2TB maximum size. The other type is the EFI label, which consists of eight partitions, labeled 0 to 8, skipping 7 and is used by default for ZFS and required for LUNs greater than 2TB.

### Scanning for new volumes

To scan for a new volume or volumes that are mapped to the Solaris system involves the use of several of the utilities described in the Useful Tools section above. After mapping the volume(s) to the Solaris host, it is necessary to rescan the Fibre Channel HBA ports so that Solaris becomes aware of the newly available device.

First, use the `fcinfo` command to determine the HBA ports to scan. For example, system fiero has a dual-port QLogic QLA2342 HBA. When running the `fcinfo` command the following output is seen:

```
[root@fiero] {/testvol1} # fcinfo hba-port
HBA Port WWN: 210000e08b17c800
  OS Device Name: /dev/cfg/c0
  Manufacturer: QLogic Corp.
  Model: QLA2342
  Firmware Version: 03.03.28
  FCode/BIOS Version: fcode: 1.17;
  Serial Number: not available
  Driver Name: qlc
  Driver Version: 20110321-3.05
  Type: N-port
  State: online
  Supported Speeds: 1Gb 2Gb
  Current Speed: 2Gb
  Node WWN: 200000e08b17c800
```

```
HBA Port WWN: 210100e08b37c800
```
Dell Compellent Storage Center Solaris Best Practices

OS Device Name: /dev/cfg/c1
Manufacturer: QLogic Corp.
Model: QLA2342
Firmware Version: 03.03.28
FCode/BIOS Version: fcode: 1.17;
Serial Number: not available
Driver Name: qlc
Driver Version: 20110321-3.05
Type: N-port
State: online
Supported Speeds: 1Gb 2Gb
Current Speed: 2Gb
Node WWN: 200100e08b37c800

We can then use the `cfgadm` command to rescan the HBA ports to discover the newly mapped volume. We run the `cfgadm` command with the `-c configure` switch to perform the configuration scan:

```
{root@fiero} {/testvol1} # cfgadm -c configure c0
{root@fiero} {/testvol1} # cfgadm -c configure c1
```

Output similar to the following is seen in the `/var/adm/messages` log:

```
Nov  6 21:06:41 fiero genunix: [ID 530209 kern.info]
/scsi_vhci/ssd@g6000d31000006700000000000000006ff (ssd0) multipath status: optimal: path 1 fp2/ssl@w5000d31000006709,0 is online: Load balancing: round-robin
Nov  6 21:06:41 fiero scsi: [ID 243001 kern.info]
/pci@0f,0/pci@1f,0/QLGC,qlc@5/fp@0,0 (fcp2):
Nov  6 21:06:41 fiero Lun=0 for target=df0300 disappeared
Nov  6 21:06:42 fiero genunix: [ID 530209 kern.info]
/scsi_vhci/ssd@g6000d310000065000000000000000060 (ssd1) multipath status: optimal: path 3 fp2/ssl@w5000d31000006513,1 is online: Load balancing: round-robin
Nov  6 21:07:15 fiero genunix: [ID 530209 kern.info]
/scsi_vhci/ssd@g6000d31000006700000000000000006ff (ssd0) multipath status: optimal: path 2 fp3/ssl@w5000d3100000670b,0 is online: Load balancing: round-robin
Nov  6 21:07:15 fiero scsi: [ID 243001 kern.info]
/pci@0f,0/pci@1f,0/QLGC,qlc@5,1/fp@0,0 (fcp3):
Nov  6 21:07:15 fiero Lun=0 for target=df0500 disappeared
Nov  6 21:07:16 fiero genunix: [ID 530209 kern.info]
/scsi_vhci/ssd@g6000d310000065000000000000000060 (ssd1) multipath status: optimal: path 4 fp3/ssl@w5000d31000006511,1 is online: Load balancing: round-robin
```

The `mpathadm` command can be used to verify both paths to the LUN are fully configured and usable:

```
{root@fiero} {/testvol1} # mpathadm list lu
/dev/rdsk/c4t6000D31000006700000000000000006FFd0s2
Total Path Count: 2
Operational Path Count: 2
```

To verify which LUN is the newly mapped LUN and not the boot-from-SAN OS LUN, the output from the `mpathadm` or `format` commands can be used to compare the device name with the Serial Number assigned to the LUN in Storage Center.

Then, use the `format` command to label the newly discovered LUN. We can see from the output below there are two LUNs reported. The OS LUN, which is a boot-from-SAN LUN from Storage Center, as well
as the newly mapped and discovered LUN are reported by `format`. Select the new LUN and label it. A default partition/slice layout will be selected using the traditional SMI label type since the LUN is less than 2TB.

```
{root@fiero} {/usr/local/bin} # format
Searching for disks...done

c4t6000D310000065000000000000006d0: configured with capacity of 100.00GB

AVAILABLE DISK SELECTIONS:
0. c4t6000D310000067000000000000006Ff0d0 <COMPELNT-CompellentVol-0504
cyl 49930 alt 2 hd 8 sec 126> SOL100S
   /scsi_vhci/ssd@g6000d310000067000000000000006ff
1. c4t6000D310000065000000000000006d0 <COMPELNT-CompellentVol-0504
cyl 49930 alt 2 hd 8 sec 525>
   /scsi_vhci/ssd@g6000d310000065000000000000006d0
Specify disk (enter its number)[0]: 1
selecting c4t6000D310000065000000000000006d0
[disk formatted]
Disk not labeled. Label it now? Yes

format> par

PARTITION MENU:
0 - change '0' partition
1 - change '1' partition
2 - change '2' partition
3 - change '3' partition
4 - change '4' partition
5 - change '5' partition
6 - change '6' partition
7 - change '7' partition
select - select a predefined table
modify - modify a predefined partition table
name - name the current table
print - display the current table
label - write partition map and label to the disk
!<cmd> - execute <cmd>, then return
quit

partition> pr
Current partition table (default):
Total disk cylinders available: 49930 + 2 (reserved cylinders)

Part  Tag   Flag  Cylinders     Size           Blocks
 0   root  wm   0 - 62  129.20MB  (63/0/0)  264600
 1   swap  wu   63 - 125 129.20MB  (63/0/0)  264600
 2   backup wu   0 - 49929 100.00GB (49930/0/0) 209706000
 3   unassigned  wu   0 - 49929 100.00GB (49930/0/0) 209706000
 4   unassigned  wu   0 - 49929 100.00GB (49930/0/0) 209706000
 5   unassigned  wu   0 - 49929 100.00GB (49930/0/0) 209706000
 6   usr   wm   126 - 49929 99.74GB  (49804/0/0) 209176800
 7   unassigned  wm   0 - 49929 99.74GB  (49804/0/0) 209176800

partition>

After exiting the `format` utility, use the `newfs` command to create a UFS file system on the LUN and mount it to the system as a usable block device.

```
{root@fiero} {/usr/local/bin} # newfs
/dev/rdsk/c4t6000D310000065000000000000006d0s6
```
newfs: construct a new file system
/dev/rdsk/c4t6000D310000065000000000000006d0s6: (y/n)? y
Warning: 1824 sector(s) in last cylinder unallocated
/dev/rdsk/c4t6000D310000065000000000000006d0s6: 209176800 sectors in
34046 cylinders of 48 tracks, 128 sectors
102137.1MB in 2128 cyl groups (16 c/g, 48.00MB/g, 5824 i/g)
super-block backups (for fsck -F ufs -o b=§) at:
32, 98464, 196896, 295328, 393760, 492192, 590624, 689056, 787488, 885920,
Initializing cylinder groups:
.........................................
super-block backups for last 10 cylinder groups at:
208208672, 208307104, 208405536, 208503968, 208602400, 208700832, 208799264,
208897696, 208996128, 209094560

{root@fiero} {/usr/local/bin} # mkdir /testvol1
{root@fiero} {/usr/local/bin} # mount
/dev/dsk/c4t6000D310000065000000000000006d0s6 /testvol1
{root@fiero} {/testvol1} # df /testvol1
File system size used avail capacity Mounted on
/dev/dsk/c4t6000D310000065000000000000006d0s6
98G 100M 97G 1% /testvol1

Correlating Storage Center serial number with OS device
There are several ways to correlate a Storage Center volume with the Solaris 10 LUN presented on the
OS host side. One way is to simply compare the device listed in format with the Serial Number listed in
the Storage Center GUI under the Volume’s General tab.

In the output of the format command, part of the device name listed contains the serial number
assigned by Storage Center. Figure 1 below highlights the LUN device identification information
presented in the format output, which can be compared to the Volume Serial Number highlighted in
Figure 2.

Figure 1. Solaris 10 Format Output
The highlighted portion above can be used to compare to the highlighted portion in the Storage Center GUI below:

![Storage Center GUI](image)

**Figure 2. Storage Center Volume Serial Number**

Another way to correlate the LUN as reported on the Solaris 10 host side to the Storage Center Serial Number is to use the `luxadm` command. The `luxadm` command has a display option that can be used to gather detailed information on the LUN. For example, the command below is run to gather details on the LUN device and this includes the Serial Number reported back by Storage Center. The highlighted portion indicates the Serial Number reported, which can be compared to the information reported in Storage Center.

```
{root@ceaser} {/} # luxadm display
/dev/rdsk/c6t6000D310000067000000000000000912d0s2
DEVICE PROPERTIES for disk: /dev/rdsk/c6t6000D310000067000000000000000913d0s2
  Vendor:           COMPELNT
  Product ID:       Compellent Vol
  Revision:         0504
  Serial Num:       00000067-00000912
  Unformatted capacity: 51200.000 MBytes
  Read Cache:       Enabled
    Minimum prefetch: 0x1
    Maximum prefetch: 0xf000
  Device Type:      Disk device
  Path(s):
    /dev/rdsk/c6t6000D310000067000000000000000913d0s2
    /devices/scsi_vhci/ssd@6000d3100000670000000000000000913:c,raw
  Controller:
    /devices/pci@7c0/pci@0/pci@8/QLGC,qlc@0/fp@0,0
    Device Address      5000d3100000671b,c8
    Host controller port WWN 2100001b3209d6da
    Class                primary
    State                ONLINE
    Controller:
```

Page 17
Removing volumes

When removing a volume/LUN from a Solaris 10 system, it is important to execute this process in the proper order to make sure the volume is cleanly removed from the system. For example, if you were to remove the mapping from the Storage Center GUI before taking the LUN offline on the host, various OS utilities would start reporting issues that can affect the system administrator's ability to manage the host effectively. Below is an example of how best to remove a volume/LUN from a Solaris 10 host that is using the Dell Compellent SAN.

The first step is to identify the LUN to be removed. This can be done in a number of ways, but for this example we will use the `format` command to identify the LUN first:

```
{root@fiero} {/} # format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
0. c4t6000d3100000067000000000000006FFd0 <COMPELNT-CompellentVol-0504
cyl 49930 alt 2 hd 8 sec 126> SOL10OS
   /sci_vhci/ssf8g6000d310000067000000000000006FF
1. c4t6000d3100000650000000000000061d0 <COMPELNT-CompellentVol-0504
cyl 49835 alt 2 hd 8 sec 263> TESTVOL2
   /sci_vhci/ssf8g6000d3100000650000000000000061
   /sci_vhci/ssf8g6000d3100000650000000000000061
2. c4t6000d3100000650000000000000060d0 <COMPELNT-CompellentVol-0504
cyl 49930 alt 2 hd 8 sec 525> TESTVOL1
   /sci_vhci/ssf8g6000d3100000650000000000000060
3. c4t6000d3100000650000000000000062d0 <COMPELNT-CompellentVol-0504-1.46TB>
   /sci_vhci/ssf8g6000d3100000650000000000000062
   /sci_vhci/ssf8g6000d3100000650000000000000062
4. c4t6000d3100000650000000000000063d0 <COMPELNT-CompellentVol-0504
cyl 49930 alt 2 hd 8 sec 525> TESTVOL4
   /sci_vhci/ssf8g6000d3100000650000000000000063
```

Specify disk (enter its number): ^C

The volume we want to remove is the one with the volume name of TESTVOL1. We can use the `luxadm` command to take the volume offline. Make sure to unmount the file system that is using the LUN before attempting to take it offline.

```
{root@fiero} {/} # umount /testvol1
{root@fiero} {/} # luxadm -e offline
/dev/rdsk/c4t6000D3100000650000000000000060d0s0
```

Then, using the Storage Center GUI, remove the mapping for the volume to the Solaris 10 host. Once that is completed, use the `cfgadm` command to rescan the fibre channel HBAs so that the controller ports no longer reference the removed volume. Remember, you can use the `fcinfo hba-port` command to get a listing of the FCP HBAs in the system.

```
{root@fiero} {/} # cfgadm -o force_update -c configure c0
{root@fiero} {/} # cfgadm -o force_update -c configure c1
```
Finally, use the `devfsadm` command to clean up the device links and references for the removed volume.

```
{root@fiero} // # devfsadm -Cv
<SNIP>
devfsadm[1287]: verbose: removing file:
/dev/dsk/c4t6000D3100000650000000000000060d0s0
devfsadm[1287]: verbose: removing file:
/dev/dsk/c4t6000D3100000650000000000000060d0s1
devfsadm[1287]: verbose: removing file:
/dev/dsk/c4t6000D3100000650000000000000060d0s2
devfsadm[1287]: verbose: removing file:
/dev/dsk/c4t6000D3100000650000000000000060d0s3
devfsadm[1287]: verbose: removing file:
/dev/dsk/c4t6000D3100000650000000000000060d0s4
devfsadm[1287]: verbose: removing file:
/dev/dsk/c4t6000D3100000650000000000000060d0s5
devfsadm[1287]: verbose: removing file:
/dev/dsk/c4t6000D3100000650000000000000060d0s6
devfsadm[1287]: verbose: removing file:
/dev/dsk/c4t6000D3100000650000000000000060d0s7
devfsadm[1287]: verbose: removing file:
```

To confirm the volume has been removed, you can review the `/var/sadm/messages` log or run the `format` or `mpathadm` commands to see the volume is no longer listed.

```
{root@fiero} // # format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
  0. c4t6000D310000067000000000000006FFd0 <COMPELNT-CompellentVol-0504
     cyl 49930 alt 2 hd 8 sec 126> SOLIOSS
     /scsi_vhci/ssd@g6000d3100000667000000000000006ff
  1. c4t6000D3100000650000000000000061d0 <COMPELNT-CompellentVol-0504
     cyl 49835 alt 2 hd 8 sec 263> TESTVOL2
     /scsi_vhci/ssd@g6000d3100000650000000000000061
  2. c4t6000D3100000650000000000000062d0 <COMPELNT-Compellent Vol-0504-
     1.46TB>
     /scsi_vhci/ssd8g6000d3100000650000000000000062
     3. c4t6000D3100000650000000000000063d0 <COMPELNT-CompellentVol-0504
     cyl 49930 alt 2 hd 8 sec 525> TESTVOL4
     /scsi_vhci/ssd8g6000d3100000650000000000000063
Specify disk (enter its number): ^C
```

```
{root@fiero} // # mpathadm list lu
/dev/rdsk/c4t6000D310000067000000000000006FFd0s2
    Total Path Count: 2
    Operational Path Count: 2
/dev/rdsk/c4t6000D3100000650000000000000061d0s2
    Total Path Count: 2
    Operational Path Count: 2
/dev/rdsk/c4t6000D3100000650000000000000062d0s2
    Total Path Count: 2
    Operational Path Count: 2
/dev/rdsk/c4t6000D3100000650000000000000063d0s2
    Total Path Count: 2
    Operational Path Count: 2
```

Dell Compellent Storage Center Solaris Best Practices

Partitions and file systems
As a block level SAN, the Compellent Storage Center will take any partition and file system scheme supported by the Solaris Operating System. However, there are some things to take into consideration when designing a scheme.

Partitions/slices
Traditionally, Solaris LUNs are provisioned with different slices (or partitions) for the function of the file system slice. For example, `/` is often on a separate slice from `/usr` or `/export/home`, etc, etc. Although this is appropriate for the primary boot volume, it is recommended to not create elaborate partition schemes when using data LUNs with Storage Center. Although multiple slices are supported, it often leads to unnecessary complexity when it comes to managing LUNs presented to Solaris 10 from Storage Center. After mapping a volume to Solaris 10, it is recommended to only use a single slice on the LUN. Dell Compellent recommends assigning volumes on a “per-purpose” basis. For example, the OS volume should be separate from a data volume (e.g. database data), and a separate volume for logging (e.g. database logs). For each volume presented, a single slice would be utilized on the volume. By allocating volumes based on a business need, the data and function of the volume can be logically administered for that purpose, which may have different retention, security, performance, etc. requirements depending on the purpose.

Disk labels
When a volume is mapped to the Solaris 10 host, subsequently discovered using `cfgadm` on the host side, and then format is executed, a disk label is then generally applied. The traditional label type was the SMI label and will be assigned by default when a LUN is less than 2TB in size. The newer label type is the EFI label. The EFI label allows for LUNs greater than 2TB in size, and is also the label type used with ZFS.

If you expect to have a UFS file system that may start below 2TB, but has the potential to grow beyond 2TB, it is recommended to use the EFI label so the UFS file system can be grown beyond the 2TB limit. In most cases it is considered a best practice to use the EFI label.

In traditional usage situations, such as the boot volume in a boot-from-SAN configuration, using an SMI label is recommended, as it is also recommended to keep the boot volume separate from data volumes, and to keep it less than 2TB.

SMI labels
A LUN that is less than 2TB will use the SMI label by default, which will setup slices using the default system defined scheme. This can be changed with the format command and limit to just slice0 to contain the whole disk, and then slice2, which is the backup location and is required to exist by the SMI label.

EFI labels
A LUN that is greater than 2TB, or has a ZFS pool created on it, will use the EFI label. The EFI label is required for ZFS and brings with it some advanced capabilities. Please refer to the Oracle documentation for further details.

UFS file system
The traditional and most common file system type on Solaris is the UFS file system. This section will talk about some recommendations when using this file system type on Solaris 10 with Dell Compellent Storage Center.
As of Solaris 10 Update 9, the UFS file system is capable of files and a file system size of 16TB. Although there may be practical limitations that would prevent such a size from being implemented in production, the Dell Compellent Storage Center supports volume sizes up to the limits of the Solaris 10 UFS implementation.

Once the volume is mapped from Storage Center to the Solaris 10 host, and the actions are taken as highlighted in the Scanning for New Volumes section above, you can put a file system on the volume and mount it for use. The `newfs` or `mkfs` commands can be used to write the file system to the volume. The example below uses the `newfs` command and then shows the volume mounted to the Solaris 10 host.

Note that in a single path configuration, the device will appear in the traditional `cNtNsN` format, but with a multipath configuration the device name appears a little different. Refer to the Multipath section below for more information. The example below is using a multipath configuration with the native MPxIO from the Solaris SFS software.

After mapping a 50GB volume and scanning for it using the `cfgadm` command, running `format` reports the discovery of the new device:

```
{root@fiero} {/} # format
Searching for disks...done

c4t6000D3100006500000000000000006d0: configured with capacity of 50.00GB

AVAILABLE DISK SELECTIONS:
  0. c4t6000D3100006500000000000000006d0 <COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 126> SOL10OS
      /scsi_vhci/ssd@g6000d3100006500000000000000006d0 <COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 126> SOL10OS
  1. c4t6000D3100006500000000000000006d0 <COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 126> TESTVOL2
      /scsi_vhci/ssd@g6000d3100006500000000000000006d0 <COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 126> TESTVOL2
  2. c4t6000D3100006500000000000000006d0 <COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 126> TESTVOL2
      /scsi_vhci/ssd@g6000d3100006500000000000000006d0 <COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 126> TESTVOL2
  3. c4t6000D3100006500000000000000006d0 <COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 126> TESTVOL2
      /scsi_vhci/ssd@g6000d3100006500000000000000006d0 <COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 126> TESTVOL2
  4. c4t6000D3100006500000000000000006d0 <COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 126> TESTVOL2
      /scsi_vhci/ssd@g6000d3100006500000000000000006d0 <COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 126> TESTVOL2

Specify disk (enter its number):
```

After labeling the device, and setting the slices to the recommended single slice, use the `newfs` command to create the UFS file system on the new volume:

```
{root@fiero} {/} # newfs
/dev/rdsk/c4t6000D3100006500000000000000006d0s0
newfs: construct a new file system
/dev/rdsk/c4t6000D3100006500000000000000006d0s0: (y/n)? y
Warning: 664 sector(s) in last cylinder unallocated
/dev/rdsk/c4t6000D3100006500000000000000006d0s0: 104852840 sectors
in 17066 cylinders of 48 tracks, 128 sectors
51197.7MB in 1067 cyl groups (16 c/g, 48.00MB/g, 5824 i/g)
super-block backups (for fsck -F ufs -o b=0) at:
  32, 98464, 196896, 295328, 393760, 492192, 590624, 689056, 787488, 885920,
Initializing cylinder groups:
```
super-block backups for last 10 cylinder groups at:
  103911584, 104010016, 104108448, 104206880, 104305312, 104403744,
  104502176, 104600608, 104699040, 104797472

Next, create the mount point and then mount the newly formatted volume for usage.

```bash
(root@fiero) {} # mkdir /testvol4
(root@fiero) {} # mount /dev/dsk/c4t6000D3100000650000000000000063d0s0 /testvol4
(root@fiero) {} # df -h /testvol4
File system size used avail capacity Mounted on
/dev/dsk/c4t6000D3100000650000000000000063d0s0 49G 50M 49G 1% /testvol4
```

To configure this for automatic mounting after system reboots, then add the following to the `/etc/vfstab` file:

```bash
/dev/dsk/c6t6000D31000006900000000000000F85d0s0 /dev/rdsk/c6t6000D31000006900000000000000F85d0s0 /testvol1 ufs 2 yes -
```

### Expanding a UFS file system

A common administration task is to allocate additional space to an existing file system. The Dell Compellent Storage Center supports the expansion of a UFS based file system using the Solaris 10 native utilities to accomplish such a task.

Below is a brief example of the process to expand a volume on the Storage Center and then grow the UFS file system to take advantage of the new space. The information below is brief, so please refer to the Dell Compellent Volume Expansion with Solaris 10 UFS technical tip for more details.

There is always some risk to data loss whenever the underlying file system is modified. Before executing any of the steps below, make sure to take a backup of the volume. Also, it is very important to keep the same partition layout. In the example below, when the LUN was first configured it was customized with a slice 0 and slice 2 only.

First, we take an existing volume/LUN with data on it and examine that it is reaching capacity. The `/testvol4` area is at 98% capacity. This space is mounted from a 50GB volume, and we want to expand it to 100GB. The process outlined below requires a short downtime in which the file system is un-mounted in order to modify the LUN’s geometry to recognize the new LUN size.

We can see the file system in question is running out of space:

```bash
(root@ceaser) {} # df -h .
Filesystem size used avail capacity Mounted on
/dev/dsk/c6t6000D31000006900000000000000F85d0s0 49G 47G 1.3G 98% /testvol4
```

Expand the volume to 100GB in the Dell Compellent Storage Center GUI.
Next, un-mount the file system and use the Solaris `format` command to configure the device using the new disk geometry. Be cautious and make sure to preserve the same partition (0-7) and starting cylinder. With Storage Center, taking a Replay of the volume prior to making any of these changes would be an option. The selection sequence is: `format` → `select device` → `type` → `auto-configure` → `label` → `p` → `p` to see new partition with added space. Then quit out of `format`.

```
{root@ceaser} {/} # umount /testvol4

{root@ceaser} {/} # format
Searching for disks...done
```
AVAILABLE DISK SELECTIONS:
0. c6t6000d31000006700000000000000008bb <COMPELNT-Compellent Vol-0504-100.00GB>
   /scsi_vhci/ssd@g6000d31000006700000000000000008bb
1. c6t6000d3100000670000000000000000074f <COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 252> SOL100S
   /scsi_vhci/ssd@g6000d3100000670000000000000000074f
2. c6t6000d3100000670000000000000000913d0 <COMPELNT-CompellentVol-0504 cyl 49835 alt 2 hd 8 sec 263>
   /scsi_vhci/ssd@g6000d31000006700000000000000000913

Specify disk (enter its number): 2
selecting c6t6000d3100000670000000000000000913d0 [disk formatted]

FORMAT MENU:

- disk: select a disk
- type: select (define) a disk type
- partition: select (define) a partition table
- current: describe the current disk
- format: format and analyze the disk
- repair: repair a defective sector
- label: write label to the disk
- analyze: surface analysis
- defect: defect list management
- backup: search for backup labels
- verify: read and display labels
- save: save new disk/partition definitions
- inquiry: show vendor, product and revision
- volname: set 8-character volume name
- !<cmd>: execute <cmd>, then return
- quit

format> type

AVAILABLE DRIVE TYPES:
0. Auto configure
1. Quantum ProDrive 80S
2. Quantum ProDrive 105S
3. CDC Wren IV 94171-344
4. SUN0104
5. SUN0207
6. SUN0327
7. SUN0340
8. SUN0424
9. SUN0535
10. SUN0669
11. SUN1.0G
12. SUN1.05
13. SUN1.3G
14. SUN2.1G
15. SUN2.9G
16. Zip 100
17. Zip 250
18. Peerless 10GB
19. COMPELNT-CompellentVol-0504
20. COMPELNT-CompellentVol-0504
21. other

Specify disk type (enter its number)[20]: 0

Specifying drive type 0
<COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 525>
selecting c6t6000d3100000670000000000000000913d0

At this point it is necessary to check the partition table. Make sure the partition layout is the same as it was when starting. Make sure the starting cylinder is correct for the data slice and modify the ending cylinder to now end at the new last cylinder. After making the changes, the slices will look like below and we can safely label the disk.

format> pa

PARTITION MENU:

<table>
<thead>
<tr>
<th>Number</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>change '0' partition</td>
</tr>
<tr>
<td>1</td>
<td>change '1' partition</td>
</tr>
<tr>
<td>2</td>
<td>change '2' partition</td>
</tr>
<tr>
<td>3</td>
<td>change '3' partition</td>
</tr>
<tr>
<td>4</td>
<td>change '4' partition</td>
</tr>
<tr>
<td>5</td>
<td>change '5' partition</td>
</tr>
<tr>
<td>6</td>
<td>change '6' partition</td>
</tr>
<tr>
<td>7</td>
<td>change '7' partition</td>
</tr>
<tr>
<td>select</td>
<td>select a predefined table</td>
</tr>
<tr>
<td>modify</td>
<td>modify a predefined partition table</td>
</tr>
<tr>
<td>name</td>
<td>name the current table</td>
</tr>
<tr>
<td>print</td>
<td>display the current table</td>
</tr>
<tr>
<td>label</td>
<td>- write partition map and label to the disk</td>
</tr>
<tr>
<td>!&lt;cmd&gt;</td>
<td>- execute &lt;cmd&gt;, then return</td>
</tr>
<tr>
<td>quit</td>
<td></td>
</tr>
</tbody>
</table>

partition> pr

Current partition table (default):  Total disk cylinders available: 49930 + 2 (reserved cylinders)

<table>
<thead>
<tr>
<th>Part</th>
<th>Tag</th>
<th>Flag</th>
<th>Cylinders</th>
<th>Size</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>root</td>
<td>wm</td>
<td>0 - 62</td>
<td>129.20MB</td>
<td>(63/0/0) 264600</td>
</tr>
<tr>
<td>1</td>
<td>swap</td>
<td>wu</td>
<td>63 - 125</td>
<td>129.20MB</td>
<td>(63/0/0) 264600</td>
</tr>
<tr>
<td>2</td>
<td>backup</td>
<td>wu</td>
<td>0 - 49929</td>
<td>100.00GB</td>
<td>(49930/0/0) 209706000</td>
</tr>
<tr>
<td>3</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>(0/0/0)</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>(0/0/0)</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>(0/0/0)</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>usr</td>
<td>wm</td>
<td>126 - 49929</td>
<td>99.74GB</td>
<td>(49804/0/0) 209176800</td>
</tr>
<tr>
<td>7</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>(0/0/0)</td>
<td>0</td>
</tr>
</tbody>
</table>

Enter partition id tag[usr]: unassigned
Enter partition permission flags[wm]:
Enter new starting cyl[126]: 0
Enter partition size[209176800b, 49804c, 102137.11mb, 99.74gb]: 0c

partition> 6

Enter partition id tag[usr]: unassigned
Enter partition permission flags[wm]:
Enter new starting cyl[126]: 0
Enter partition size[209176800b, 49804c, 102137.11mb, 99.74gb]: 0c

partition> 1

Enter partition id tag[swap]: unassigned
Enter partition permission flags[wm]:
Enter new starting cyl[63]: 0
Enter partition size[264600b, 63c, 62e, 129.20mb, 0.13gb]: 0c

partition> 0

Enter partition id tag[root]:
Enter partition permission flags[wm]:
Enter new starting cyl[0]: 0
Enter partition size [264600b, 63c, 62e, 129.20mb, 0.13gb]: 49930c

Current partition table (unnamed):
Total disk cylinders available: 49930 + 2 (reserved cylinders)

<table>
<thead>
<tr>
<th>Part</th>
<th>Tag</th>
<th>Flag</th>
<th>Cylinders</th>
<th>Size</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>root</td>
<td>wm</td>
<td>0 - 49929</td>
<td>100.00GB</td>
<td>(49930/0/0) 209706000</td>
</tr>
<tr>
<td>1</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>2</td>
<td>backup</td>
<td>wu</td>
<td>0 - 49929</td>
<td>100.00GB</td>
<td>(49930/0/0) 209706000</td>
</tr>
<tr>
<td>3</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>4</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>5</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>6</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>7</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td></td>
<td>(0/0/0) 0</td>
</tr>
</tbody>
</table>

Quit out of the format utility and re-mount the file system:

format> q

The old file system size is still reported:

```
{root@ceaser} {/testvol4} # df -h .
Filesystem size used avail capacity Mounted on
/dev/dsk/c6t6000D31000006700000000000000913d0s0 49G 47G 1.3G 98% /testvol4
```

Perform a `growfs` on the file system to take advantage of the new volume size. During this process, the LUN is fully available. At the end we can see the new size:

```
{root@ceaser} {/testvol4} # growfs -M /testvol4
/dev/rdsk/c6t6000D31000006700000000000000913d0s0 Warning: 1008 sector(s) in last cylinder unallocated
/dev/rdsk/c6t6000D31000006700000000000000913d0s0: 209706000
sectors in 34132 cylinders of 48 tracks, 128 sectors
102395.5MB in 2134 cyl groups (16 c/g, 48.00MB/g, 5824 i/g)
```
super-block backups (for `fsck -F ufs -o b=`) at:
  32, 98464, 196896, 295328, 393760, 492192, 590624, 689056, 787488, 885920,
Initializing cylinder groups:
..........................................
super-block backups for last 10 cylinder groups at:
  208799264, 208897696, 208996128, 209094560, 209192992, 209291424,
  209389856, 209488288, 209586720, 209685152

{root@ceaser} {/testvol4} # df -h .
Filesystem size used avail capacity Mounted on
/dev/dsk/c6t6000D310000006700000000000000913d0s0 98G 47G 51G 49% /testvol4

UFS and Data Instant Replay
Taking a Replay of a UFS based volume is relatively straightforward. The Replay is an exact
time copy of the UFS file system at the time the Replay was taken. Therefore, it is
important to understand how this “state” needs to be considered when later using the contents
of the Replay. Below are some general best practice guidelines and considerations to be aware
of when working with Replays of UFS volumes.

1. It is recommended to map the View volume created from a Replay to a different
   system than the system in which the Replay was taken. If attempting to recover file(s)
it is recommended to map the View volume to a “recovery” server and then copy the
   necessary data back to the original system over the network. However, it is supported
to map the View volume to the original system, but care needs to be taken to not
   confuse the Production volume with the View volume. For example, if volume names
   (set by the `format` command’s VOLNAME option) are used, when the View volume is
   mapped it will present itself with the same VOLNAME and this could lead to confusion.
2. If the UFS file system is mounted when the Replay is taken, all mount flags are
   preserved in the Replay image. When the Replay volume is then presented to another
   server, those UFS mount flags are still intact.
3. When mounting the Replay of a UFS file system that was mounted at the time the
   Replay was taken, the system administrator may be prompted that a file system
   consistency check is required. This is normal, and should be run using the standard
   Solaris `fsck` program.

Below is an example of the process to follow to take a Replay of a UFS based volume, create a
view volume of the Replay, and present it to a recover host (also running Solaris 10), recover
some files and then remove the View volume. In this example, a volume has regular replays
taken daily at 12:01AM. A user has accidentally removed a file that needs to be recovered.

The “testvol2” mount is a UFS file system that contains hundreds of files. A user has
accidentally removed the files ending in 3, 4 and 5. These need to be recovered:

{root@fiero} {/testvol2} # ls
 1     testfile.16    testfile.26    testfile.36    testfile.46
  testfile.56    testfile.66    testfile.76    testfile.87    testfile.97
lost+found/    testfile.17    testfile.27    testfile.37    testfile.47
  testfile.57    testfile.67    testfile.77    testfile.88    testfile.98
  testfile.0     testfile.18    testfile.28    testfile.38    testfile.48
  testfile.58    testfile.68    testfile.78    testfile.89    testfile.99
To recover these files, we need to first identify the replay that has the file needed for recovery. This can be accomplished by reviewing the “Replays” tab for the volume in question.

Next, create a “View Volume” from the selected Replay. This will create a new volume, with its own unique serial number, containing the point-in-time blocks frozen when that Replay was taken. This volume can be mapped to the source system or a recovery system following the process discussed in the Scanning for New Volumes section above.

Right-click on the Replay in the list, and then select the “Create volume from Replay” option to create the View volume.
Figure 6. **Create View Volume**

Map this new volume to a Solaris 10 system (or any system that can read a UFS file system type). Once mapped, mount the file system, run `fsck` (if prompted) and then recover the file(s) needed. In this case we will map this new volume to another Solaris 10 system called `vito`, which is our recovery server.

Figure 7. **Map View Volume Summary Screen**
Using the `cfgadm` command, discover the View volume on the recovery system, create a mount point, run `fsck` (if needed) and mount the device.

```
{root@vito} {/} # cfgadm -o force_update -c configure c0
{root@vito} {/} # cfgadm -o force_update -c configure c1

{root@vito} {/} # df -h /
Filesystem size used avail capacity Mounted on
/dev/dsk/c4t6000D3100006700000000000700d0s0 22G 11G 11G 51% /

{root@vito} {/} # mpathadm list lu
/dev/rdsk/c4t6000D3100006700000000000700d0s2 Total Path Count: 2
Operational Path Count: 2
/dev/rdsk/c4t6000D3100006500000000000006Ad0s2 Total Path Count: 2
Operational Path Count: 2

{root@vito} {/} # mkdir /recovery
{root@vito} {/} # mkdir /recovery/fiero-testvol2

{root@vito} {/} # mount /dev/dsk/c4t6000D3100006500000000000006Ad0s0
/recovery/fiero-testvol2/

{root@vito} {/recovery/fiero-testvol2} # ls
lost+found/ testfile.0 testfile.1 testfile.2 testfile.3
testfile.4 testfile.5 testfile.6 testfile.7 testfile.9

{root@vito} {/recovery/fiero-testvol2} # scp testfile.3 testfile.4
testfile.5 fiero:/testvol2/
Password:
testfile.3  100% | ********************* | 100 MB  00:15

{root@vito} {/recovery/fiero-testvol2} # ll testfile.3 testfile.4
testfile.5  100% | ********************* | 100 MB  00:15

{root@fiero} {/testvol2} # ll testfile.3 testfile.4 testfile.5
-rw------- 1 root root 100M Nov 14 14:40 testfile.3
-rw------- 1 root root 100M Nov 14 14:40 testfile.4
-rw------- 1 root root 100M Nov 14 14:40 testfile.5
```

The files have now been recovered. We can unmount and remove the View volume from the recovery server. This process can also be scripted, and ultimately automated, with the Dell Compellent Command Utility (CompCU) which is discussed further in the Scripting/Automation section later in this paper.

**UFS performance tuning**

Performance tuning is a bit of a black-art when it comes to file systems. The UFS file system is no different. The performance of a file system is very highly dependent on the actual IO workload occurring on the said file system. Much consideration must be taken when discussing ways to improve IO performance on a file system. This paper does not go into any great detail with regard to performance tuning of the UFS file system on Solaris 10. There are a number of Oracle as well as Solaris community guides that can be leveraged that discuss performance considerations in more detail. In particular to the UFS file system the following is a good place to review: [http://www.solarisinternals.com/si/reading/fs2/fs2.html](http://www.solarisinternals.com/si/reading/fs2/fs2.html). The primary point to understand is that an understanding of the workload occurring on the file system is necessary in order to begin to understand what type of tuning may best benefit the IO performance.
maxcontig
One of the tunable parameters to the UFS file system that can result in performance improvement is the `maxcontig` setting. This value controls the number of file system blocks that will be read or written in a single operation. The value is set automatically by the Solaris 10 OS when the file system is created and is based on the device characteristics. Adjusting this value can result in improved performance when it is set to a value that is optimal to the IO workload.

To see the current `maxcontig` value on the file system use the `fstyp` command as exampled below:

```
{root@fiero} {/testvol4} # fstyp -v
/dev/dsk/c4t6000D310000065000000000000000063d0s0 | grep maxcontig
maxcontig 128  rtdelay 0ms  rps  250
```

You can also check the `maxcontig` value with the `mkfs` command with the “-m” option as exampled below:

```
{root@fiero} {/} # mkfs -m
/dev/rdsk/c4t6000D310000065000000000000000063d0s0
mkfs -F ufs -o
  nsect=128,ntrack=48,bsize=8192,fragsize=1024,cgsize=16,free=1,rps=250,nbp
  i=8155,opt=t,apc=0,gap=0,nrpos=8,maxcontig=128,mtb=n
  /dev/rdsk/c4t6000D310000065000000000000000063d0s0 209706000
```

To change the `maxcontig` setting use the `tunefs` command. An example that changes it from the current, auto-set value of 128 to 256:

```
{root@fiero} {/} # tunefs -a 256
/dev/dsk/c4t6000D310000065000000000000000063d0s2
maximum contiguous block count changes from 128 to 256
```

```
{root@fiero} {/testvol4} # fstyp -v
/dev/dsk/c4t6000D310000065000000000000000063d0s0 | grep maxcontig
maxcontig 256  rtdelay 0ms  rps  250
```

Direct I/O
This refers to I/O that does not traffic through the OS page cache. By default none of the file systems on Solaris, including UFS, enable Direct I/O. However, some workloads can greatly benefit from using Direct I/O so enabling this can result in significant performance increases. However, keep in mind that when enabling Direct I/O, file system features such as read pre-fetching, write coalescing, and staged writes will not exist, which can lead to poor performance.

For example, if the Solaris 10 system will be holding an Oracle RDBMS instance, using Direct I/O can lead to increase in performance. Since Oracle uses its own caching mechanism within its own SGA (Shared Global Area), read operations using global OS cache can lead to double-cache situations. In addition, there is a higher CPU load when OS cache is heavily used, so in situations in which high I/O rate is occurring, the OS locking mechanisms used to keep cache coherency can introduce a “speed limit” that is far below what the underlying Storage Center SAN is capable of producing.
Below are some reasons that UFS Direct I/O can improve performance. This information is pooled from: [http://www.solarisinternals.com/wiki/index.php/Direct_I/O](http://www.solarisinternals.com/wiki/index.php/Direct_I/O)

1. It eliminates the write-breakup of large synchronous writes, which are important for the log-writer.
2. It allows concurrent read/writes to a single file (eliminates the POSIX reader/writer lock)
3. It eliminates double buffering
4. It provides a small, efficient code path for reads and writes
5. It removes memory pressure from the operating environment. Solaris uses the virtual memory system to implement file system caching and, as a result, many page-in operations result when reading data though the file system cache. With UFS direct I/O, database reads do not need to involve the virtual memory system.

The UFS Direct I/O option can be enabled on a per file system basis by adding the `-o forcedirectio` option to the `mount` command:

```bash
{root@fiero} {/} # mount -o forcedirectio /dev/dsk/c4t6000d3100006500000000000063d0s0 /testvol4
```

```bash
{root@fiero} {/testvol4} # mount | grep testvol4
/testvol4 on /dev/dsk/c4t6000d310000650000000000000063d0s0 read/write/setuid/devices/rstchown/intr/forcedirectio/largefiles/logging/xattr/onerror=panic/dev=1d80030 on Fri Nov 11 13:21:16 2011
```

Whether or not a workload will benefit from Direct I/O is something that should be thoroughly tested before implementing into a production environment.

For example, on a Solaris 10 host having a 50GB volume mapped and formatted with default UFS settings, creating a 4GB file with the `mkfile` command took 73 seconds, which is an average throughput of 56MB/sec. Remounting that same volume with the Direct I/O option resulted in an average throughput of 11MB/sec. Although these values are not indicative of how well the Dell Compellent Storage Center SAN can perform, the singular change in adding/removing the UFS Direct I/O option can have a dramatic affect on performance, depending on the I/O workload.

/etc/system

The `etc/system` file contains system wide kernel parameters. This file is read at boot time, therefore any changes made to this file will require a reboot to be enabled. There are a couple of exceptions to this, but generally rebooting the server is recommended when modifying this file. The Solaris tuning guide found at the following URL covers this: [http://download.oracle.com/docs/cd/E19253-01/817-0404/](http://download.oracle.com/docs/cd/E19253-01/817-0404/).

An example of a change that can affect IO performance is to change the maximum SCSI transfer size. To enable this change, add an entry such as below to the `/etc/system` file and reboot the server to activate.

* Allow larger SCSI I/O transfers, parameter is bytes
ZFS file system
ZFS is a combined file system as well as a logical volume manager developed by Sun. It is natively included in Solaris 10 Update 6 and newer releases. ZFS brings with it the ability to create Exabyte (EB) size files and file system. The stated maximums for ZFS are 16 EB for a maximum file size as well as 16 EB for maximum volume size. A ZFS file system is capable of containing a maximum of $2^{48}$ files. ZFS has a number of advanced file system features such as built-in compression, encryption, deduplication as well as others. Not all features are available depending on the actual version being used. In addition, some of the features of ZFS are redundant to features available in Storage Center. This section will discuss some of the Best Practices that should be considered when using ZFS on Solaris 10 systems utilizing the Dell Compellent Storage Center SAN. For some greater detail on ZFS, as well as general ZFS tuning and best practices, refer to the following URLs to get started:


ZFS and Storage Center volumes
The Dell Compellent SAN stripes and mirrors all data across all drives in the SAN. Therefore, it is recommended to create large, single volumes for each zpool created rather than creating multiple, smaller volumes.

In addition, adding multiple zpools, each with a dedicated large volume, balanced across multiple Storage Center controllers is recommended when trying to achieve optimal IO performance.

Mapping a volume for ZFS
The process to map a volume for ZFS usage is identical to the process above, with just a minor variation available. It is recommended to use the entire volume to contain the ZFS zpool rather than create multiple partitions/slices. A disk label is still required, and ZFS requires the use of the EFI label type.

First, map the volume to the Solaris 10 host as described above. Once mapped, perform a scan of the FCP HBA port(s) so the Solaris 10 host can discover the newly mapped volume. Running format will show the new volume is listed.

```bash
{root@fiero} {/} # format
Searching for disks...done

c4t6000D310000650000000000000064d0: configured with capacity of 100.00GB

AVAILABLE DISK SELECTIONS:
  0. c4t6000D310000670000000000000006FD0 <COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 126> SOL10OS
/scsi_vhci/ssd@g6000d31000067000000000000006ff
```
1. `c4t6000d3100000650000000000000064d0 <COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 525> /scsi_vhci/ssd@g6000d3100000650000000000000064d0`<br>2. `c4t6000d3100000650000000000000063d0 <COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 525> TESTVOL4 /scsi_vhci/ssd@g6000d3100000650000000000000063d0`<br>Specify disk (enter its number):

There is no need to label the device at this point, simply create a zpool on the device. This will automatically label the device with an EFI label:

```
{root@fiero} {/} # zpool create cmlzpool c4t6000D310000065000000000000064d0
```

```
{root@fiero} {/} # format
Searching for disks...done
```

**AVAILABLE DISK SELECTIONS:**

0. `c4t6000D310000067000000000000006FFd0 <COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 126> SOL100S /scsi_vhci/ssd@g6000d310000067000000000000006ff`

1. `c4t6000D3100000650000000000000064d0 <COMPELNT-CompellentVol-0504-100.00GB> /scsi_vhci/ssd@g6000d3100000650000000000000064`

2. `c4t6000D3100000650000000000000063d0 <COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 525> TESTVOL4 /scsi_vhci/ssd@g6000d3100000650000000000000063`

Specify disk (enter its number): 1

```
<SNIP>
```

```
partition> pr
Current partition table (original):
Total disk sectors available: 209698782 + 16384 (reserved sectors)
```

```
<table>
<thead>
<tr>
<th>Part</th>
<th>Tag</th>
<th>Flag</th>
<th>First Sector</th>
<th>Size</th>
<th>Last Sector</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>usr</td>
<td>wm</td>
<td>256</td>
<td>99.99GB</td>
<td>209698782</td>
</tr>
<tr>
<td>1</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>reserved</td>
<td>wm</td>
<td>209698783</td>
<td>8.00MB</td>
<td>209715166</td>
</tr>
</tbody>
</table>
```

```
{root@fiero} {/} # zpool list
NAME      SIZE  ALLOC FREE  CAP  HEALTH  ALTROOT
---------  ------- ------- ------ ---- ------- --------
cmlzpool  99.5G 168K  99.5G 0%  ONLINE -
```

You can now create a ZFS file system within the pool and begin using the space.

```
{root@fiero} {/} # zfs create cmlzpool/zfsdata
```

```
{root@fiero} {/} # zfs list
NAME     USED  AVAIL REFER MOUNTPOINT
--------- ------- ------- ------ ------------
cmlzpool 134K  97.9G  32K    /cmlzpool
cmlzpool/zfsdata 31K  97.9G  31K    /cmlzpool/zfsdata
```
Expanding a ZFS file system

There are a couple of ways to expand a ZFS file system on a Solaris 10 system using the Dell Compellent SAN. The current implementation of ZFS on Solaris 10 (as of this writing) does not support the online expansion of a zpool device that has had its underlying volume expanded in size on the Dell Compellent Storage Center SAN. Therefore, simply performing a LUN Expansion on the Storage Center and then expanding the file system on the host side does not work. There are a couple of options available to expand the available storage to an existing ZFS pool. These are talked about below.

There is always some risk to data loss whenever the underlying file system is modified. Before executing any of the steps below, make sure to take a backup of the volume.

zpool add

The first option for expanding a ZFS volume involves simply creating a new volume on Storage Center, mapping it to the Solaris 10 host and then adding the new volume device to the existing ZFS pool.

In this example, the current zpool is at 73% capacity. We want to expand its capacity by adding another 100GB to the existing 100GB available to the zpool. We first create a new 100GB volume in Storage Center and then map it to the Solaris 10 host. Following the familiar steps above, the volume is discovered and can be seen with `format/mpathadm/etc`.

```
{root@fiero} {/cmlzpool/zfsdata} # zpool list
NAME       SIZE  ALLOC   FREE    CAP  HEALTH  ALTROOT
cmlzpool  99.5G  73.2G  26.3G    73%  ONLINE
```

```
{root@fiero} {/cmlzpool/zfsdata} # zpool list
NAME       SIZE  ALLOC   FREE    CAP  HEALTH  ALTROOT
cmlzpool  199G  73.2G   126G    36%  ONLINE
```

With the above approach it should be pointed out that with the device added to the existing pool it is basically going to concatenate to the new device. ZFS will equally use free space on the original device until that space is consumed, but by default there is no restripe of existing data. Keep in mind, though, that Storage Center always stripes data across all drives.

Once the space is consumed on one of the volumes, all I/O will occur to the remaining volume(s) with free space. Therefore, when using multiple volumes in a single zpool, it is recommended to balance the volumes equally across both Storage Center controllers. This will help optimize performance as I/O load can be spread across both controllers.

zpool replace

Another option for allocating more space and expanding an existing ZFS file system is to create a new volume on Storage Center of the size we want the ZFS file system to have. You can then perform a zpool replace operation which will migrate the data from the original volume to the new volume. This has the advantage of keeping the number of volumes to manage to a fewer number. The disadvantage here is that if the existing file system has a large amount of data, it can take some time for the data migration to complete. Below is an example of this process.

A 100GB ZFS pool is mounted to /cmlpool/data1. It is running out of space, and is currently at 81% utilization. The system administrator wants to create a 250GB volume and replace the
existing 100GB volume with the new 200GB volume. The advantage to this approach is that this process can be done with the file system online and data accessible throughout the process.

We have an existing 100GB zpool that is at 81% of capacity.

```
{root@ceaser} {/cmlpool/data1} # df -h /cmlpool/data1/
Filesystem size used avail capacity Mounted on
/cmlpool/data1 98G 79G 19G 81% /cmlpool/data1
```

A new 250GB volume is created and mapped to the Solaris 10 host. The normal steps to discover the newly mapped volume are run and the `format` command displays the new LUN that is discovered:

```
{root@ceaser} {/cmlpool/data1} # format
Searching for disks...done
c6t6000D3100000650000000000000006Cd0: configured with capacity of 249.99GB
```

AVAILABLE DISK SELECTIONS:
0. c6t6000D3100000650000000000000006Bd0 <COMPELNT-Compellent Vol-0504-100.00GB>
   /scsi_vhci/ssd@g6000d3100000650000000000000006Bd0
1. c6t6000D3100000650000000000000006Cd0 <COMPELNT-Compellent Vol-0504 cyl 63748 alt 2 hd 32 sec 257>
   /scsi_vhci/ssd@g6000d3100000650000000000000006Cd0
2. c6t6000D31000006700000000000000074Fd0 <COMPELNT-Compellent Vol-0504 cyl 49930 alt 2 hd 8 sec 252>
   /scsi_vhci/ssd@g6000d3100000670000000000000074Fd0

Specify disk (enter its number):

We can now perform a zpool replace that will replace the original zpool device (c6t6000D3100000650000000000000006Bd0) with the new zpool device (c6t6000D3100000650000000000000006Cd0):

```
{root@ceaser} {/cmlpool/data1} # zpool replace cmlpool
c6t6000D3100000650000000000000006Bd0
c6t6000D3100000650000000000000006Cd0
```

The additional space is not yet available:

```
{root@ceaser} {/cmlpool/data1} # zpool list
NAME SIZE ALLOC FREE CAP HEALTH ALTROOT
/cmlpool 99.5G 77.9G 21.6G 78% ONLINE -
```

We can run a `zpool status <pool name>` on the pool to check the progress of the replace migration:

```
{root@ceaser} {/cmlpool/data1} # zpool status cmlpool
pool: cmlpool
state: ONLINE
status: One or more devices is currently being resilvered. The pool will continue to function, possibly in a degraded state.
action: Wait for the resilver to complete.
scan: resilver in progress since Thu Mar 16 22:52:09 2000
  5.91G scanned out of 77.9G at 116M/s, 0h10m to go
  5.91G scanned out of 77.9G at 116M/s, 0h10m to go
  5.90G resilvered, 7.58% done
```
config:

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATE</th>
<th>READ</th>
<th>WRITE</th>
<th>CKSUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmlpool</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>replacing-0</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>c6t6000D31000006500000000000000006Bd0</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>c6t6000D31000006500000000000000006Cd0</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

errors: No known data errors

During the resilvering process, new data can be written. However, performance is expectedly degraded due to the additional I/O overhead occurring.

A typical looking `iostat -x 1` output that shows the read/write I/O occurring as the data is being migrated from the old device to the new device:

```
{root@ceaser} {/} # iostat -x 1
extended device statistics
device  r/s  w/s  kr/s  kw/s  wait  actv  svc_t  %w  %b
sd2     0.0  0.0  0.0   0.0   0.0  0.0   0.0   0   0
ssd0    0.0  0.0  0.0   0.0   0.0  0.0   0.0   0   0
ssd2    973.1 0.0 121383.0 0.0   0.0   9.6    9.9  1 99
ssd3    0.0  970.1 0.0 121379.5 0.0   7.1   7.4  1 95
nfs1     0.0   0.0  0.0   0.0   0.0  0.0   0.0   0   0
extended device statistics
device  r/s  w/s  kr/s  kw/s  wait  actv  svc_t  %w  %b
sd2     0.0  0.0  0.0   0.0   0.0  0.0   0.0   0   0
ssd0    0.0  0.0  0.0   0.0   0.0  0.0   0.0   0   0
ssd2    976.0 0.0 122895.7 0.0   0.0   9.2    9.5  1100
ssd3    0.0  972.0 0.0 122129.8 0.0   7.6   7.8  1 96
nfs1     0.0   0.0  0.0   0.0   0.0  0.0   0.0   0   0
extended device statistics
device  r/s  w/s  kr/s  kw/s  wait  actv  svc_t  %w  %b
sd2     0.0  0.0  0.0   0.0   0.0  0.0   0.0   0   0
ssd0    0.0  0.0  0.0   0.0   0.0  0.0   0.0   0   0
ssd2    957.1 0.0 120470.7 0.0   0.0   9.0    9.4  1 98
ssd3    0.0  950.1 0.0 119321.0 0.0   7.0   7.4  1 97
nfs1     0.0   0.0  0.0   0.0   0.0  0.0   0.0   0   0
```

After the resilvering process completes, a zpool status output will look similar to the following:

```
{root@ceaser} {/cmlpool/data1} # zpool status cmlpool
pool: cmlpool
state: ONLINE
scan: resilvered 77.9G in 0h11m with 0 errors on Thu Mar 16 23:03:41 2000
config:

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATE</th>
<th>READ</th>
<th>WRITE</th>
<th>CKSUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmlpool</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>c6t6000D31000006500000000000000006Cd0</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

errors: No known data errors

Starting with Solaris 10 Update 9, the ZFS software was updated to include the ability to expand the zpool size online. On an existing pool, this can be done with the command:

```
zpool online -e <pool> <device>
```
This command causes the disk geometry to be re-read, thus making the newly allocated space available:

```bash
{root@ceaser} {/cmlpool/data1} # zpool list
NAME      SIZE  ALLOC   FREE    CAP  HEALTH  ALTROOT
---------  ------  ------  ------  ----  -------  -------
cmlpool   99.5G  78.9G  20.6G    79%  ONLINE  -
```

```bash
{root@ceaser} {/cmlpool/data1} # zpool online -e cmlpool
c6t6000D31000006500000000000000006Cd0
{root@ceaser} {/cmlpool/data1} # zpool list
NAME      SIZE  ALLOC   FREE    CAP  HEALTH  ALTROOT
---------  ------  ------  ------  ----  -------  -------
cmlpool   250G  78.9G  171G    31%  ONLINE  -
```

**ZFS and Data Instant Replay**

Data Instant Replay (DIR) is a core feature of the Dell Compellent Storage Center SAN and its usage can dramatically improve efficiency when it comes to facilitating backup/recovery tasks, sharing production data with other environments (e.g. development, system test, QA, etc). However, due to the manner in which ZFS stores metadata pertaining to the zpool, it is not possible to point a View Volume of a ZFS file system back to the source system while the live zpool is active, thus complicating the usage of DIR with ZFS.

As with the UFS file system, the recommended practice is to map View Volumes based on Replays to a different host than to the source host. However, where it is possible to mount a UFS based View Volume back to the source host, but care should be taken if this is done, when ZFS is used, this is not possible due to the metadata ZFS stores on each disk device that makes up the zpool.

When using multiple Dell Compellent Storage Center volumes to make a ZFS pool, be sure to configure these volumes to use a Consistency Group. This will ensure that all volumes in the zpool are successfully snapshotted with Data Instant Replay at the same point-in-time state as to keep the zpool state consistent. Refer to the Storage Center User Guide for more details on consistency groups.

**Mapping a Replay of a ZFS File System**

Below is a brief example of the process to map a Replay taken of a ZFS file system to a recovery server. The main difference here with a UFS based file system is that when the ZFS pool is imported, there will be an OS complaint that the zpool was previously active on a different system. The ‘-f’ switch will have to be added to the command to force the import to complete.

First, the output below shows that several files are “accidentally” deleted from the source device.

```bash
{root@ceaser} {/cmlpool/data1} # ll testfile.9*
-rw-------T 1 root root 100M Mar 16 22:25 testfile.9
-rw-------T 1 root root 100M Mar 16 22:27 testfile.90
-rw-------T 1 root root 100M Mar 16 22:27 testfile.91
-rw-------T 1 root root 100M Mar 16 22:27 testfile.92
-rw-------T 1 root root 100M Mar 16 22:27 testfile.93
-rw-------T 1 root root 100M Mar 16 22:27 testfile.94
-rw-------T 1 root root 100M Mar 16 22:27 testfile.95
-rw-------T 1 root root 100M Mar 16 22:27 testfile.96
-rw-------T 1 root root 100M Mar 16 22:27 testfile.97
```
Use the Storage Center GUI to create a View Volume of the most recent Replay that contains the files mistakenly deleted above. Map this View Volume to the recovery server (fiero).

After scanning for the new volume, the format command will show the View Volume in the output:

```
{root@fiero} {/} # format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
0. c4t6000d31000000650000000000000006Dd0 <COMPELNT-Compellent Vol-0504-500.00GB> ➜ ZFS View Volume
   /scsi_vhci/ssd@g6000d31000000650000000000000006Dd0
1. c4t6000d31000000670000000000000006FFd0 <COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 126> SOL100S
   /scsi_vhci/ssd@g6000d31000000670000000000000006FF
2. c4t6000d310000006500000000000000068d0 <COMPELNT-Compellent Vol-0504-100.00GB>
   /scsi_vhci/ssd@g6000d310000006500000000000000068
3. c4t6000d310000006500000000000000069d0 <COMPELNT-CompellentVol-0504 cyl 49835 alt 2 hd 8 sec 263>
   /scsi_vhci/ssd@g6000d310000006500000000000000069

Specify disk (enter its number): ^C
```

Attempting to import the ZFS pool directly will result in the following warning message:

```
{root@fiero} {/} # zpool import cmlpool
cannot import 'cmlpool': pool may be in use from other system, it was last accessed by ceaser (hostid: 0x83029572) on Thu Mar 16 23:19:05 2000 use '-f' to import anyway
```

Simply use the ‘-f’ option to import the zpool:

```
{root@fiero} {/} # zpool import -f cmlpool
{root@fiero} {/} # zpool list
NAME     SIZE    ALLOC   FREE    CAP    HEALTH ALTROOT
cmlpool   250G    78.9G   171G    31%     ONLINE -
cmlzpool  99.5G   73.2G   26.3G    73%     ONLINE -
{root@fiero} {/} # zpool status cmlpool
   pool: cmlpool
   state: ONLINE
   scan: resilvered 77.9G in 0h11m with 0 errors on Thu Mar 16 23:03:41 2000
   config:
   NAME     STATE      READ    WRITE   CKS
   cmlpool   ONLINE    0       0       0
   c4t6000d31000000650000000000000006Dd0 ONLINE    0       0       0

errors: No known data errors
```
We can now recover the deleted files back to the original Solaris 10 system:

```
{root@fiero} {/} # cd /cmlpool/data1/
{root@fiero} {/cmlpool/data1} # ll testfile.9*
-rw------- 1 root  root  100M Mar 16 2000 testfile.9
-rw------- 1 root  root  100M Mar 16 2000 testfile.90
-rw------- 1 root  root  100M Mar 16 2000 testfile.91
-rw------- 1 root  root  100M Mar 16 2000 testfile.92
-rw------- 1 root  root  100M Mar 16 2000 testfile.93
-rw------- 1 root  root  100M Mar 16 2000 testfile.94
-rw------- 1 root  root  100M Mar 16 2000 testfile.95
-rw------- 1 root  root  100M Mar 16 2000 testfile.96
-rw------- 1 root  root  100M Mar 16 2000 testfile.97
-rw------- 1 root  root  100M Mar 16 2000 testfile.98
-rw------- 1 root  root  100M Mar 16 2000 testfile.99

{root@fiero} {/cmlpool/data1} # scp testfile.9* ceaser:/cmlpool/data1/
```

The files are recovered and once again available on the source (production) system.

```
{root@ceaser} {/cmlpool/data1} # ll testfile.9*
-rw------- 1 root  root  100M Mar 17 19:53 testfile.9
-rw------- 1 root  root  100M Mar 17 19:53 testfile.90
-rw------- 1 root  root  100M Mar 17 19:53 testfile.91
-rw------- 1 root  root  100M Mar 17 19:53 testfile.92
-rw------- 1 root  root  100M Mar 17 19:54 testfile.93
-rw------- 1 root  root  100M Mar 17 19:54 testfile.94
-rw------- 1 root  root  100M Mar 17 19:54 testfile.95
-rw------- 1 root  root  100M Mar 17 19:55 testfile.96
-rw------- 1 root  root  100M Mar 17 19:55 testfile.97
-rw------- 1 root  root  100M Mar 17 19:55 testfile.98
-rw------- 1 root  root  100M Mar 17 19:55 testfile.99
```

The files have now been recovered. We can unmount and remove the View volume from the recovery server. This process can also be scripted, and ultimately automated, with the Dell Compellent Command Utility (CompCU), which is discussed further in the Scripting/Automation section later in this paper.

**ZFS performance tuning**

ZFS does an incredibly good job of basically auto-tuning itself and using its built-in algorithms to provide optimal performance. There are several places in which advanced performance tuning of ZFS is discussed in greater detail, one such location is:


Some points to consider from the perspective of ZFS on Dell Compellent Storage Center is discussed below. These are general points of interest, as such; any modifications made should be thoroughly tested before implementing in production. Also, before making any changes to the ZFS device, verify a good backup of the data exists.

**ZFS Intent Log (ZIL)**

Traditionally, it has been recommended with other storage platforms/designs to dedicate spindles for ZIL. With Storage Center it is not necessary to store ZIL on separate storage. Storage Center by default writes all new data to Tier 1 (fastest) storage in the SAN. In addition all data is striped across all drives in the tier. Therefore, placing ZIL data on separate LUNs
does not lead to a performance increase. Allowing Storage Center to automatically manage the best placement for new data is recommended.

Cache Flush
Storage Center will always return acknowledgements from writes to stable storage.

Multiple Storage Center Volumes in ZFS Pool
There can be some performance benefit when configuring multiple Storage Center volumes balanced across both Storage Center controllers. By using multiple volumes, there is more I/O path available by leveraging both controllers. However, care should be taken to not set Fibre Channel timeouts too low. If these are set too low and a controller failure was to occur, the volume(s) on that controller have to move to the other (up) controller. If this does not complete in the timeout period, SCSI errors are reported up the stack to ZFS and will cause the pool to go offline.

Multipath with Solaris 10
The Solaris 10 Operating System includes built-in multipath I/O support beginning with Update 2. As mentioned earlier in this paper, Dell Compellent believes that Oracle, as the Solaris 10 OS developer, is best suited to develop the I/O connection, or stack, into its servers. Therefore, it is recommended to use the built-in Solaris SAN Foundation Suite software stack when connecting Solaris 10 hosts to Dell Compellent Storage Center SANs.

By leveraging the built-in OS solution, there is no additional software required in order to connect a Solaris 10 system to a Storage Center based SAN. In addition, when OS updates are made available that include the SFS software, the SAN connectivity suite is updated as well. Additional knowledge and training to learn a 3rd party stack to manage connectivity is not required, which also simplifies the connectivity between the server and SAN devices.

This section will talk about some of the tasks required to activate multipath on a Solaris 10 system using the built-in SFS suite which includes the Sun Traffic Manager and MPxIO components. Recommended settings for path timeouts are also discussed.

Configuring and Activating Multipath (MPxIO) on Solaris 10 with Storage Center
After initially installing the Solaris 10 operating system, multipath will not be activated automatically. It is necessary to perform a few tasks in order to activate the multipath configuration using the Solaris SFS, which will utilize the MPxIO stack. The steps below outline the tasks to complete to properly activate MPxIO for use with the Dell Compellent SAN.

1. If needed, add the second HBA port path. Complete the fabric zone configuration as appropriate for the environment.
2. Run the `cfgadm` command to discover the additional path to the LUN, again if needed.
3. Run the `format` command and two devices will now appear in the output. These are the two device paths back to the same LUN on the Storage Center SAN.

```bash
{root@fiero} {/} # format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
0. c0t5000D31000006709d0 <COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 126> SOL10OS
   /pci@1f,0/pci@1/QLGC,qlc@5/fp@0,0/ssd@w5000d31000006709,0
```
1. `c1t5000D3100000670Bd0 <COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 126> SOL100OS /pci@1f,0/pci@1/QLGC,qlc@5,1/fp@0,0/ssd@w5000d3100000670b,0`

4. Modify the `/kernel/drv/scsi_vhci.conf` file by adding the Dell Compellent device definition to the end of the file. The device definition is exampled below, and must be entered exactly as shown as the spacing and syntax is very specific:

   ```
   device-type-scsi-options-list = "COMPELNTCompellent Vol", "symmetric-option";
   symmetric-option = 0x1000000;
   ```

5. Run the Sun Traffic Manager program (`stmsboot`) to enable multipath (MPxIO). You are prompted to reboot the system in order to activate MPxIO. During this process the `/etc/vfstab` file and the dump configuration will be updated to reflect the device changes.

   ```
   {root@fiero} {/} # stmsboot -e
   WARNING: stmsboot operates on each supported multipath-capable controller
detected in a host. In your system, these controllers are

   /pci@1f,0/pci@1/QLGC,qlc@5/fp@0,0
   /pci@1f,0/pci@1/QLGC,qlc@5,1/fp@0,0

   If you do NOT wish to operate on these controllers, please quit stmsboot
and re-invoke with `-D { fp | mpt | mpt_sas} to specify which
controllers you wish to modify your multipathing configuration for.

Do you wish to continue? [y/n] (default: y)
WARNING: This operation will require a reboot.
Do you want to continue ? [y/n] (default: y)
The changes will come into effect after rebooting the system.
Reboot the system now ? [y/n] (default: y)
Nov 18 10:41:59 fiero reboot: rebooted by root
   updating /platform/sun4u/boot_archive
<SNIP>
stmsboot: configuring devices
stmsboot: vfstab has been updated
   Dump content: kernel pages
   Dump device:
   /dev/dsk/c4t6000D3100000670000000000000006FFd0s1 (dedicated)
Savecore directory: /var/crash/fiero
   Savecore enabled: yes
   Save compressed: on
stmsboot: dump configuration has been updated.
stmsboot: now regenerating boot archive
   The / file system
   (/dev/rdsk/c4t6000D31000006700000000000000006FFd0s0) is being .

6. After the system is back online, login and run the `format` command again. You will now see just one device in the output. The device will now include part of the device designation for the particular Storage Center SAN.

   ```
   {root@fiero} {/} # format
   Searching for disks...done
   ```
Matching the MPxIO Device Name with /dev/rdsk Device Entry
If there is a need to correlate the MPxIO device name to the original physical device path name, the `stmsboot` command can be used for this purpose. Login to the Solaris 10 system as root and run the following:

```bash
{root@ceaser} {/} # stmsboot -L
non-STMS device name                    STMS device name
------------------------------------------
/dev/rdsk/c1t5000D31000006709d0 /dev/rdsk/c6t6000D3100000670000000000000074Fd0
/dev/rdsk/c2t5000D3100000670Bd0 /dev/rdsk/c6t6000D3100000670000000000000074Fd0
```

Dell Compellent Storage Center Controller Failover Timing
The Dell Compellent Storage Center can take up to 60 seconds to complete volume failover from one controller to the other. The actual amount of time will vary depending on the number of volumes that need to be moved. Careful consideration should be taken when setting failover timing values on the Solaris 10 host. Setting the value too low can result in SCSI errors being reported up the stack, thus causing file systems, such as ZFS pools, to go offline in an attempt to protect data integrity.

Unfortunately, the default kernel port timer values are too low to ensure that Storage Center controller failover completes before errors are reported. Therefore, it is necessary to tweak some of the kernel values to allow sufficient time for failovers to complete.

The key value is the `fcp_offline_delay`, which is set to 20 seconds by default. This is too low, and thus needs to be adjusted up to the 60 second Dell Compellent recommendation.

To check the current value of this setting use the `adb -kw` command. The example below illustrates this usage:

```bash
{root@ceaser} {/} # adb -kw
physmem f71e2
fcp_offline_delay /D
fcp_offline_delay:          20
```

To change this value, open the `/kernel/drv/fcp.conf` file and add the following line to the end of the file:

```
fcp_offline_delay=60;
```

The system will need to be rebooted for the change to be applied. After rebooting, run the same commands as above to verify the value has been modified:

```bash
{root@ceaser} {/} # adb -kw
physmem f71e2
fcp_offline_delay /D
fcp_offline_delay:          60
```
In single path environments, the 60 second recommendation should be strictly followed. In multipath environments some tuning can be done on a per-configuration basis to lower this value, if desired. Again, however, setting this too low can result in LUNs going offline and file systems being placed in a “dirty” state by Solaris. Considerable testing should be done in which all possible failure scenarios are tested before lowering the values and using in production.

If it is desired by the system administrator to try and lower the failover timings for specific workload, application or policy needs, the following values can be lowered on the Solaris 10 system.

The following information is for experimental purposes, but is not recommended for general usage for failover timings.

The value to modify is the `fcp_offline_delay` again. Set this to the desired value in the `/kernel/drv/fcp.conf` file.

Assuming a QLogic HBA is in use and is using the `qlc` driver, then the `port-down-retry-count` and `login-retry-count` values in the `/kernel/drv/qlc.conf` file can also be modified to lower values.

Finally, the `fp_offline_ticker` can be lowered to experiment with shorter timeout settings. By default this is set to 90 seconds, which means that a path failure is not actually fully reported up the stack for as much as 150 seconds as the `fcp_offline_delay` and `fp_offline_ticker` values are honored. To reduce the time from the default of 90 seconds, add a line similar to the following to the `/kernel/drv/fcp.conf` file:

```
fp_offline_ticker=XX;
```

Software iSCSI

The Dell Compellent Storage Center supports both Fibre Channel and iSCSI SAN connectivity. The features of Storage Center are not locked into either protocol. Although one connectivity protocol may be more appropriate than the other depending on the situation, Storage Center features are available and supported on both types of connectivity.

This section covers some of the basics regarding Software iSCSI connectivity between a Solaris 10 host and a Dell Compellent Storage Center system. The information below is intended as a starting point for utilizing software iSCSI, but is not all-encompassing, but rather equips the system administrator with enough information to get started with Solaris 10 iSCSI with Storage Center.

Further information on iSCSI administration with Solaris 10 can be found on the Oracle site: [http://download.oracle.com/docs/cd/E19082-01/819-2723/fmvd/index.html](http://download.oracle.com/docs/cd/E19082-01/819-2723/fmvd/index.html)

Beginning with Solaris 10 Update 2, the onboard Ethernet ports can be used to initiate an iSCSI connection. Several packages need to be installed in order to use Software iSCSI on Solaris 10. First, verify the necessary packages are installed on the Solaris 10 host:
The Solaris 10 iSCSI target software can use standard volume(s) from the Dell Compellent Storage Center as a backing store. All the Storage Center needs to provide is a standard volume to the server via normal device discovery methods, using built-in Solaris 10 commands and utilities (e.g. iscsiadm, devfsadm, format, etc). No additional software is required.

**Mapping an iSCSI volume to a Solaris 10 Host**

Verify the iSCSI control port IP addresses on the Storage Center controllers. As with Fibre Channel connectivity, multipath configuration is recommended. Best practice recommendation is to have at least two control ports configured on Storage Center, and to have these on two different subnets within the network.

```
{root@ceaser} {/root} # pkginfo SUNWiscsiu SUNWiscsir
system       SUNWiscsiu Sun iSCSI Management Utilities (usr)

The Solaris 10 iSCSI target software can use standard volume(s) from the Dell Compellent Storage Center as a backing store. All the Storage Center needs to provide is a standard volume to the server via normal device discovery methods, using built-in Solaris 10 commands and utilities (e.g. iscsiadm, devfsadm, format, etc). No additional software is required.

**Mapping an iSCSI volume to a Solaris 10 Host**

Verify the iSCSI control port IP addresses on the Storage Center controllers. As with Fibre Channel connectivity, multipath configuration is recommended. Best practice recommendation is to have at least two control ports configured on Storage Center, and to have these on two different subnets within the network.

```
{root@ceaser} {/root} # pkginfo SUNWiscsiu SUNWiscsir
system       SUNWiscsiu Sun iSCSI Device Driver (root)

The Solaris 10 iSCSI target software can use standard volume(s) from the Dell Compellent Storage Center as a backing store. All the Storage Center needs to provide is a standard volume to the server via normal device discovery methods, using built-in Solaris 10 commands and utilities (e.g. iscsiadm, devfsadm, format, etc). No additional software is required.

**Mapping an iSCSI volume to a Solaris 10 Host**

Verify the iSCSI control port IP addresses on the Storage Center controllers. As with Fibre Channel connectivity, multipath configuration is recommended. Best practice recommendation is to have at least two control ports configured on Storage Center, and to have these on two different subnets within the network.

```
{root@ceaser} {/root} # pkginfo SUNWiscsiu SUNWiscsir
system       SUNWiscsiu Sun iSCSI Management Utilities (usr)

The Solaris 10 iSCSI target software can use standard volume(s) from the Dell Compellent Storage Center as a backing store. All the Storage Center needs to provide is a standard volume to the server via normal device discovery methods, using built-in Solaris 10 commands and utilities (e.g. iscsiadm, devfsadm, format, etc). No additional software is required.

**Mapping an iSCSI volume to a Solaris 10 Host**

Verify the iSCSI control port IP addresses on the Storage Center controllers. As with Fibre Channel connectivity, multipath configuration is recommended. Best practice recommendation is to have at least two control ports configured on Storage Center, and to have these on two different subnets within the network.

```
{root@ceaser} {/root} # pkginfo SUNWiscsiu SUNWiscsir
system       SUNWiscsiu Sun iSCSI Device Driver (root)

The Solaris 10 iSCSI target software can use standard volume(s) from the Dell Compellent Storage Center as a backing store. All the Storage Center needs to provide is a standard volume to the server via normal device discovery methods, using built-in Solaris 10 commands and utilities (e.g. iscsiadm, devfsadm, format, etc). No additional software is required.

**Mapping an iSCSI volume to a Solaris 10 Host**

Verify the iSCSI control port IP addresses on the Storage Center controllers. As with Fibre Channel connectivity, multipath configuration is recommended. Best practice recommendation is to have at least two control ports configured on Storage Center, and to have these on two different subnets within the network.

```
{root@ceaser} {/root} # pkginfo SUNWiscsiu SUNWiscsir
system       SUNWiscsiu Sun iSCSI Management Utilities (usr)

The Solaris 10 iSCSI target software can use standard volume(s) from the Dell Compellent Storage Center as a backing store. All the Storage Center needs to provide is a standard volume to the server via normal device discovery methods, using built-in Solaris 10 commands and utilities (e.g. iscsiadm, devfsadm, format, etc). No additional software is required.

**Mapping an iSCSI volume to a Solaris 10 Host**

Verify the iSCSI control port IP addresses on the Storage Center controllers. As with Fibre Channel connectivity, multipath configuration is recommended. Best practice recommendation is to have at least two control ports configured on Storage Center, and to have these on two different subnets within the network.

```
{root@ceaser} {/root} # pkginfo SUNWiscsiu SUNWiscsir
system       SUNWiscsiu Sun iSCSI Management Utilities (usr)

The Solaris 10 iSCSI target software can use standard volume(s) from the Dell Compellent Storage Center as a backing store. All the Storage Center needs to provide is a standard volume to the server via normal device discovery methods, using built-in Solaris 10 commands and utilities (e.g. iscsiadm, devfsadm, format, etc). No additional software is required.

**Mapping an iSCSI volume to a Solaris 10 Host**

Verify the iSCSI control port IP addresses on the Storage Center controllers. As with Fibre Channel connectivity, multipath configuration is recommended. Best practice recommendation is to have at least two control ports configured on Storage Center, and to have these on two different subnets within the network.

```
{root@ceaser} {/root} # pkginfo SUNWiscsiu SUNWiscsir
system       SUNWiscsiu Sun iSCSI Management Utilities (usr)

The Solaris 10 iSCSI target software can use standard volume(s) from the Dell Compellent Storage Center as a backing store. All the Storage Center needs to provide is a standard volume to the server via normal device discovery methods, using built-in Solaris 10 commands and utilities (e.g. iscsiadm, devfsadm, format, etc). No additional software is required.

**Mapping an iSCSI volume to a Solaris 10 Host**

Verify the iSCSI control port IP addresses on the Storage Center controllers. As with Fibre Channel connectivity, multipath configuration is recommended. Best practice recommendation is to have at least two control ports configured on Storage Center, and to have these on two different subnets within the network.

```
{root@ceaser} {/root} # pkginfo SUNWiscsiu SUNWiscsir
system       SUNWiscsiu Sun iSCSI Management Utilities (usr)

The Solaris 10 iSCSI target software can use standard volume(s) from the Dell Compellent Storage Center as a backing store. All the Storage Center needs to provide is a standard volume to the server via normal device discovery methods, using built-in Solaris 10 commands and utilities (e.g. iscsiadm, devfsadm, format, etc). No additional software is required.

**Mapping an iSCSI volume to a Solaris 10 Host**

Verify the iSCSI control port IP addresses on the Storage Center controllers. As with Fibre Channel connectivity, multipath configuration is recommended. Best practice recommendation is to have at least two control ports configured on Storage Center, and to have these on two different subnets within the network.

```
{root@ceaser} {/root} # pkginfo SUNWiscsiu SUNWiscsir
system       SUNWiscsiu Sun iSCSI Management Utilities (usr)

The Solaris 10 iSCSI target software can use standard volume(s) from the Dell Compellent Storage Center as a backing store. All the Storage Center needs to provide is a standard volume to the server via normal device discovery methods, using built-in Solaris 10 commands and utilities (e.g. iscsiadm, devfsadm, format, etc). No additional software is required.

**Mapping an iSCSI volume to a Solaris 10 Host**

Verify the iSCSI control port IP addresses on the Storage Center controllers. As with Fibre Channel connectivity, multipath configuration is recommended. Best practice recommendation is to have at least two control ports configured on Storage Center, and to have these on two different subnets within the network.

```
{root@ceaser} {/root} # pkginfo SUNWiscsiu SUNWiscsir
system       SUNWiscsiu Sun iSCSI Management Utilities (usr)

The Solaris 10 iSCSI target software can use standard volume(s) from the Dell Compellent Storage Center as a backing store. All the Storage Center needs to provide is a standard volume to the server via normal device discovery methods, using built-in Solaris 10 commands and utilities (e.g. iscsiadm, devfsadm, format, etc). No additional software is required.

**Mapping an iSCSI volume to a Solaris 10 Host**

Verify the iSCSI control port IP addresses on the Storage Center controllers. As with Fibre Channel connectivity, multipath configuration is recommended. Best practice recommendation is to have at least two control ports configured on Storage Center, and to have these on two different subnets within the network.

```
{root@ceaser} {/root} # pkginfo SUNWiscsiu SUNWiscsir
system       SUNWiscsiu Sun iSCSI Management Utilities (usr)

The Solaris 10 iSCSI target software can use standard volume(s) from the Dell Compellent Storage Center as a backing store. All the Storage Center needs to provide is a standard volume to the server via normal device discovery methods, using built-in Solaris 10 commands and utilities (e.g. iscsiadm, devfsadm, format, etc). No additional software is required.

**Mapping an iSCSI volume to a Solaris 10 Host**

Verify the iSCSI control port IP addresses on the Storage Center controllers. As with Fibre Channel connectivity, multipath configuration is recommended. Best practice recommendation is to have at least two control ports configured on Storage Center, and to have these on two different subnets within the network.

```
{root@ceaser} {/root} # pkginfo SUNWiscsiu SUNWiscsir
system       SUNWiscsiu Sun iSCSI Management Utilities (usr)

The Solaris 10 iSCSI target software can use standard volume(s) from the Dell Compellent Storage Center as a backing store. All the Storage Center needs to provide is a standard volume to the server via normal device discovery methods, using built-in Solaris 10 commands and utilities (e.g. iscsiadm, devfsadm, format, etc). No additional software is required.

**Mapping an iSCSI volume to a Solaris 10 Host**

Verify the iSCSI control port IP addresses on the Storage Center controllers. As with Fibre Channel connectivity, multipath configuration is recommended. Best practice recommendation is to have at least two control ports configured on Storage Center, and to have these on two different subnets within the network.

```
{root@ceaser} {/root} # pkginfo SUNWiscsiu SUNWiscsir
system       SUNWiscsiu Sun iSCSI Management Utilities (usr)

The Solaris 10 iSCSI target software can use standard volume(s) from the Dell Compellent Storage Center as a backing store. All the Storage Center needs to provide is a standard volume to the server via normal device discovery methods, using built-in Solaris 10 commands and utilities (e.g. iscsiadm, devfsadm, format, etc). No additional software is required.

**Mapping an iSCSI volume to a Solaris 10 Host**

Verify the iSCSI control port IP addresses on the Storage Center controllers. As with Fibre Channel connectivity, multipath configuration is recommended. Best practice recommendation is to have at least two control ports configured on Storage Center, and to have these on two different subnets within the network.
172.16.26.4 is alive

{root@ceaser} {/root} # ping 10.10.140.4
10.10.140.4 is alive

In addition, Jumbo frames (MTU Size) can be enabled to help improve performance over the iSCSI Ethernet network. To enable this, update the /kernel/drv/<device>.conf file. In the example system used for this paper, the iSCSI interfaces used the e1000g NIC type, therefore, the configuration file to update was the /kernel/drv/e1000g.conf file.

For example, in the /kernel/drv/e1000g.conf file change the line referring to the Maximum Frame Size from:

```
MaxFrameSize=0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
```

To:

```
MaxFrameSize=3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3;
```

After making this change, reboot the Solaris 10 host.

Before the changes, `ifconfig` would report:

```
{root@ceaser} {/kernel/drv} # ifconfig -a
e100g0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
   inet 172.16.26.82 netmask fffff000 broadcast 172.16.31.255
      ether 0:14:4f:2:95:72

e100g1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
   inet 10.10.26.99 netmask ff000000 broadcast 10.255.255.255
      ether 0:14:4f:2:95:73

e100g2: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 4
   inet 172.16.26.99 netmask fffff000 broadcast 172.16.31.255
      ether 0:14:4f:2:95:74
```

Now, after making the changes and rebooting, the interfaces now support up to 16K Frame Sizes and report the higher supported MTU size and can be seen with the `ifconfig` command:

```
{root@ceaser} {/} # ifconfig -a
e100g0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 9216 index 2
   inet 172.16.26.82 netmask fffff000 broadcast 172.16.31.255
      ether 0:14:4f:2:95:72

e100g1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 9216 index 3
   inet 10.10.26.99 netmask ff000000 broadcast 10.255.255.255
      ether 0:14:4f:2:95:73

e100g2: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 9216 index 4
   inet 172.16.26.99 netmask fffff000 broadcast 172.16.31.255
      ether 0:14:4f:2:95:74
```

To complete the process to enable Jumbo Frames, make the necessary configuration changes on the attached Ethernet switch(es). Refer to the switch vendor for details on how enable Jumbo Frames on that device.
Use the `iscsiadm` command to setup the discovery address to each iSCSI control port on the Storage Center controller(s). Then, enable the `sendtargets` discovery method. This can take several seconds to complete.

```
{root@ceaser} {/root} # iscsiadm add discovery-address 172.16.26.4:3260
{root@ceaser} {/root} # iscsiadm add discovery-address 10.10.140.4:3260

{root@ceaser} {/} # iscsiadm list discovery-address
Discovery Address: 172.16.26.4:3260
Discovery Address: 10.10.140.4:3260

{root@ceaser} {/root} # iscsiadm modify discovery --sendtargets enable
```

Verify the `sendtargets` option is enabled:

```
{root@ceaser} {/root} # iscsiadm list discovery
Discovery:
  Static: disabled
  Send Targets: enabled
  iSNS: disabled
```

Next, create a server object in Storage Center that contains the iSCSI ports created for the Solaris 10 host. Notice that a single iSCSI entry is reported, but that both IP addresses are listed for the Solaris 10 hosts IQN.
Figure 9. Example Server Object Creation - iSCSI IQN Port Information

Next, assign a server name and the appropriate Operating System type.
Finally, review the attributes associated with the new server object, then click on the **Create Now** button.
The server object associated with the iSCSI ports on the Solaris 10 host is now defined and available in the Storage Center definition. A volume can be created and mapped to this server object. In the example below a 100GB volume is created and mapped to the new object just created.

Create the Solaris 10 device links for the iSCSI HBAs using the `devfsadm` command. After the device links are created, you can use the `format` command to see the new volume that was mapped.

```
{root@ceaser} {/root} # devfsadm -i iscsi -v
{root@ceaser} {/} # dmesg | tail
Mar 23 18:16:34 ceaser scsi: [ID 583861 kern.info] ssd19 at scsi_vhci0: unit-address g6000d31000069000000000000000000f85: g6000d31000069000000000000000000f85
```
Mar 23 18:16:34 ceaser genunix: [ID 936769 kern.info] ssd19 is /scsi_vhci/ssd@g6000d3100000690000000000000f85
Mar 23 18:16:34 ceaser genunix: [ID 408114 kern.info] /scsi_vhci/ssd@g6000d3100000690000000000000f85 (ssd19) online
Mar 23 18:16:34 ceaser genunix: [ID 483743 kern.info] /scsi_vhci/ssd@g6000d3100000690000000000000f85 (ssd19) multipath status: degraded: path 5 iscsi0/ssd@0000iqn.2002-03.com.compellent:5000d310000069230000,1 is online
Mar 23 18:16:34 ceaser genunix: [ID 530209 kern.info] /scsi_vhci/ssd@g6000d3100000690000000000000f85 (ssd19) multipath status: optimal: path 6 iscsi0/ssd@0000iqn.2002-03.com.compellent:5000d310000069220000,1 is online: Load balancing: round-robin

{root@ceaser} {/root} # format
Searching for disks...done
c6t6000D3100000690000000000000F85d0: configured with capacity of 100.00GB

AVAILABLE DISK SELECTIONS:
0. c6t6000D3100000690000000000000F85d0 <COMPELNT=CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 525> This can be compared to the volume SN on Storage Center as exampled with Fibre Channel in Figure 2.
1. c6t6000D31000006700000000000000074Fd0 <COMPELNT=CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 252> SOL100S

Specify disk (enter its number):

The device can be partitioned (if desired) and labeled using the format command as desired:

{root@ceaser} {/root} # format
Searching for disks...done
c6t6000D3100000690000000000000F85d0: configured with capacity of 100.00GB

AVAILABLE DISK SELECTIONS:
0. c6t6000D3100000690000000000000F85d0 <COMPELNT=CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 525>
1. c6t6000D31000006700000000000000074Fd0 <COMPELNT=CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 252> SOL100S

Specify disk (enter its number): 0
selecting c6t6000D3100000690000000000000F85d0 [disk formatted]
Disk not labeled. Label it now? yes

<SNIP>

<Modify Partition(s), if desired>

Current partition table (unnamed):
Total disk cylinders available: 49930 + 2 (reserved cylinders)

<table>
<thead>
<tr>
<th>Part</th>
<th>Tag</th>
<th>Flag</th>
<th>Cylinders</th>
<th>Size</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>root</td>
<td>wm</td>
<td>0 - 49929</td>
<td>100.00GB</td>
<td>(49930/0/0) 209706000</td>
</tr>
<tr>
<td>1</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>2</td>
<td>backup</td>
<td>wu</td>
<td>0 - 49929</td>
<td>100.00GB</td>
<td>(49930/0/0) 209706000</td>
</tr>
<tr>
<td>3</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>4</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>5</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
</tbody>
</table>
A file system can be created on the device and mounted as needed:

```bash
{root@ceaser} [/root] # newfs /dev/rdsk/c6t6000D31000006900000000000000F85d0s0
newfs: construct a new file system
/dev/rdsk/c6t6000D31000006900000000000000F85d0s0: (y/n)? y
Warning: 1008 sector(s) in last cylinder unallocated
/dev/rdsk/c6t6000D31000006900000000000000F85d0s0: 209706000 sectors in
34132 cylinders of 48 tracks, 128 sectors
102395.5MB in 2134 cyl groups (16 c/g, 48.00MB/g, 5824 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
32, 98464, 196936, 295328, 393760, 492192, 590624, 689056, 787488, 885920,
Initializing cylinder groups:
..........................................
super-block backups for last 10 cylinder groups at:
208799264, 208897696, 208996128, 209094560, 209192992, 209291424, 209389856,
209488288, 209586720, 209685152

{root@ceaser} [/root] # mkdir /iscsi-testvol1
{root@ceaser} [/root] # mount /dev/dsk/c6t6000D31000006900000000000000F85d0s0
/iscsi-testvol1/

{root@ceaser} [/root] # cd /iscsi-testvol1/
{root@ceaser} [/iscsi-testvol1] # df -h.
Filesystem size used avail capacity Mounted on
/dev/dsk/c6t6000D31000006900000000000000F85d0s0 98G 100M 97G 1% /iscsi-testvol1

In addition, the `mpathadm` command can be used to verify that both paths configured with the iSCSI device are indeed active.

```bash
{root@ceaser} [/root] # mpathadm list 1u
/dev/rdsk/c6t6000D3100000670000000000000074Fd0s2
Total Path Count: 2
Operational Path Count: 2
/dev/rdsk/c6t6000D31000006900000000000000F85d0s2
Total Path Count: 2
Operational Path Count: 2
```

**Remove an iSCSI Volume from a Solaris 10 Host**

The process to remove a Dell Compellent Storage Center volume via iSCSI from a Solaris 10 host is very similar to the process with Fibre Channel. The obvious difference is the commands used within the Solaris 10 OS to remove, cleanup and refresh the device tree. Again, the `iscsiadm` is the primary command to be utilized to accomplish this task.

However, early releases of Solaris 10 contained a bug that prevented the complete removal of a previously attached iSCSI volume without a system reboot. To ensure that an iSCSI volume can be removed without the need for a disruptive, system-wide reboot, patch 125388-04 or higher should be installed. This patch can be installed individually, along with any dependent patches, or as part of the latest 10_Recommended patch cluster.
In the walk-through below, the iSCSI volume mapped above is removed illustrating the process required to remove an iSCSI connected volume.

First, verify the Solaris 10 system is properly patched:

```
root@ceaser} {/iscsi-testvol1} # showrev -p | grep 125388-04
Patch: 146232-08 Obsoletes: 119090-33, 121980-03, 123500-02, 125388-04, 141878-12, 143645-16 Requires: 118833-36, 119042-09, 121901-01, 125081-13, 142909-04
```

Incompatibles: Packages: SUNWiscsir, SUNWiscsiu, SUNWmpathadm

Unmount the file system using the iSCSI volume if necessary:

```
[root@ceaser] [/] # umount /iscsi-testvol1/
```

Use the `luxadm` command to offline the LUN:

```
[root@ceaser] [/] # luxadm -e offline
/dev/rdsk/c6t6000D31000006900000000000000F85d0s0
```

Remove the mappings for the volume to the Solaris 10 host in the Storage Center GUI. Note: CompCU could also be used for this purpose, see the Scripting/Automation with CompCU section later in this paper.

Use the `devfsadm` command to clean up the device entries on the Solaris 10 host:

```
[root@ceaser] [/] # devfsadm -C
```
Remove an iSCSI Discovery Address from a Solaris 10 Host

If the iSCSI IP address on the Storage Center control port were to change, it becomes necessary to remove the old address from the Solaris 10 host. Once again, the `iscsiadm` command is used to make this modification.

The configuration starts with two discovery addresses.

```
[root@ceaser] {} # iscsiadm list discovery-address
Discovery Address: 172.16.26.4:3260
Discovery Address: 10.10.140.4:3260
```

With both discovery addresses, the configuration is in a multipath state and both paths are active:

```
[root@ceaser] {}[iscsi-testvol1] # mpathadm list lu
/dev/rdsk/c6t6000D31000006700000000000000074Fd0s2
  Total Path Count: 2
  Operational Path Count: 2
/dev/rdsk/c6t6000D310000069000000000000000F85d0s2
  Total Path Count: 2
  Operational Path Count: 2
```

One of the discovery addresses is removed. The `iscsiadm` list function reports just the one active discovery address.

```
[root@ceaser] {}[iscsi-testvol1] # iscsiadm remove discovery-address
172.16.26.4:3260
```

```
[root@ceaser] {}[iscsi-testvol1] # iscsiadm list discovery-address
Discovery Address: 10.10.140.4:3260
```

The `mpathadm` command reports that only one path is now available.

```
[root@ceaser] {}[iscsi-testvol1] # mpathadm list lu
/dev/rdsk/c6t6000D31000006700000000000000074Fd0s2
  Total Path Count: 2
  Operational Path Count: 2
/dev/rdsk/c6t6000D310000069000000000000000F85d0s2
  Total Path Count: 1
  Operational Path Count: 1
```
Multiple iSCSI Paths (MPxIO) vs. Single Path iSCSI Performance Comparison

It is interesting to note that with only a single iSCSI path, performance is expectedly reduced. The following output shows one hundred 100MB files created with a single iSCSI path, including the amount of time it took to create the files, indicating an average transfer rate of 63.69MB/second.

```
{root@ceaser} {/iscsi-testvol1} # time mkfiles.ksh 100 199 100m
<SNIP>
Creating file:  testfile.195
Creating file:  testfile.196
Creating file:  testfile.197
Creating file:  testfile.198
Creating file:  testfile.199

real    2m36.506s
user    0m3.380s
sys     1m40.653s
```

After adding a second path to the iSCSI configuration, performance is increased as illustrated by the output below. Also, the Solaris 10 iSCSI target stack works natively with the MPxIO stack to automatically discover the new path to the iSCSI volume. With two iSCSI paths (using onboard 1GbE ports), throughput of one hundred 100MB files is reported as: 86.96MB/second.

```
{root@ceaser} {/iscsi-testvol1} # iscsiadm add discovery-address 172.16.26.4:3260
{root@ceaser} {/iscsi-testvol1} # iscsiadm list discovery-address
Discovery Address: 10.10.140.4:3260
Discovery Address: 172.16.26.4:3260
{root@ceaser} {/iscsi-testvol1} # mpathadm list lu
/dev/rdsk/c6t6000D3100000670000000000000000074Fd0s2
    Total Path Count: 2
    Operational Path Count: 2
/dev/rdsk/c6t6000D31000006900000000000000000F85d0s2
    Total Path Count: 2
    Operational Path Count: 2

{root@ceaser} {/iscsi-testvol1} # time mkfiles.ksh 200 299 100m
<SNIP>
Creating file:  testfile.296
Creating file:  testfile.297
Creating file:  testfile.298
Creating file:  testfile.299

real    1m54.805s
user    0m3.383s
sys     1m40.763s
```

In addition to the redundancy provided with multipathed iSCSI connectivity, there can be performance gains with multiple paths.

Scripting/Automation with CompCU

The Dell Compellent Storage Center SAN can have many of its daily functions managed through a remote command utility call the Dell Compellent Command Utility (CompCU). This allows for scripting and automation integration of SAN tasks between the Solaris 10 operating system and Storage Center. CompCU is a java-based application and thus requires the installation of java on the Solaris 10 host.
Dell Compellent Storage Center Solaris Best Practices

CompCU can be used to script common administrative tasks that can tremendous time savers and provide a consistent framework for managing volumes and replays.

To use CompCU, the server must have the proper java release installed. Refer to the Command Utility User Guide for more details. The CompCU.jar object can be downloaded from the Dell Compellent support site. Once installed on the Solaris 10 server, this tool can be used to perform Storage Center tasks from the shell prompt, which can be incorporated into new or existing end-user management scripts. Below are some common use cases for using CompCU.

- Creating Volumes, mapping to the server.
- Taking replays, recovering replays, etc.

By no means do the examples below cover the full breadth of the usefulness of CompCU, the examples below are designed to just give an initial insight into the sorts of tasks that can be automated with CompCU. In addition, the examples below are run on a Solaris 10 Update 10 system that is connected to a Storage Center SAN in a multipathed fibre channel configuration. The server object uses both ports of an Emulex LP12004-M8 and QLogic QLE2462 HBAs.

Verifying Java and initially configuring and testing CompCU functions

The first task is to install the necessary Java version onto the Solaris 10 system. Depending on the package selection during Solaris 10 installation, java may already be installed with a compatible version.

```bash
[root@ceaser] {/} # which java
/usr/bin/java
[root@ceaser] {/} # java -version
java version "1.6.0_26"
Java(TM) SE Runtime Environment (build 1.6.0_26-b03)
Java HotSpot(TM) Server VM (build 20.1-b02, mixed mode)
```

Download the CompCU package from the Dell Compellent support site. The package will include a PDF user guide as well as the CompCU.jar file needed. Place this in a logical area to be reference when using the utility. Verify CompCU is working with java by executing the command below to get the ‘help’ output displayed:

```bash
[root@ceaser] {/root/demo} # java -jar CompCU.jar -h
Compellent Command Utility (CompCU) 5.5.1.4
```

usage: java -jar CompCU.jar [Options] "<Command>"
-c <arg> Run a single command (option must be within quotes)
-default Saves host, user, and password to encrypted file
-defaultname <arg> File name to save default host, user, and password encrypted file to
-file <arg> Save output to a file
-h Show help message
-host <arg> IP Address/Host Name of Storage Center Management IP
-password <arg> Password of user
-s <arg> Run a script of commands
-user <arg> User to log into Storage Center with
-verbose Show debug output
-xmloutputfile <arg> File name to save the CompCU return code in xml format. Default is cu_output.xml.
To facilitate the ease of access to using CompCU, you can run CompCU with the “-default” switch to initially configure an encrypted password file. Below is an example of this command syntax to use:

```
{root@ceaser} {/root/demo} # java -jar compcu.jar -default -host sc9 -user Admin -password XXX
Compellent Command Utility (CompCU) 5.5.1.4
===============================================================================
User Name: Admin
Host/IP Address: sc9
===============================================================================
Connecting to Storage Center: sc9 with user: Admin
Saving CompCu Defaults to file [default.cli]...
{root@ceaser} {/root/demo} # ll default.cli
-rw-r--r-- 1 root root 32 Mar 19 16:17 default.cli
{root@ceaser} {/root/demo} # cat default.cli
P,`Yekÿ)HrEðt'0Æ56û}o
```

This will create a default file called default.cli. You can simply rename this file to match the Storage Center it refers to and reference it in future commands.

This file can then be referenced in other commands to login to the Storage Center and perform the requested actions. The example below shows a volume listing from the Storage Center named “sc9”.

```
{root@ceaser} {/root/demo} # java -jar compcu.jar -defaultname sc9-pwdfile.cli -c "volume show"
Compellent Command Utility (CompCU) 5.5.1.4
===============================================================================
User Name: Admin
Host/IP Address: sc9
Single Command: volume show
===============================================================================
Connecting to Storage Center: sc9 with user: Admin
Running Command: volume show
Index | Name | Status | ConfigSize | ActiveSize | ReplaySize | Folder | StorageProfile | DeviceID | SerialNumber | ConfigSizeBlock | ActiveSizeBlock | ReplaySizeBlock | MaxWriteSizeBlock | ReadCache | WriteCache
-----|------|--------|------------|------------|------------|--------|----------------|---------|--------------|----------------|----------------|----------------|------------------|-----------|-----------------|
 108 | ceaser-testvol1-qlc | Up | 100.00 GB | 10.24 GB | 0.00 KB | Solaris | Recommended | 6000d3100000650000000000000006e 00000065-0000006e 209715200 | 21471232 | 0 | 0 | Enabled | Enabled |
 109 | ceaser-testvol2-emlx | Up | 100.00 GB | 10.23 GB | 0.00 KB | Solaris | | | | |
```

---

Page 57
Using CompCU to automate common tasks
This section will illustrate some example use-cases with managing common Storage Center tasks with CompCU on Solaris 10. As mentioned above, these examples are indicative of the types of tasks that can be easily accomplished from the Solaris 10 shell prompt using CompCU. These are meant as starting points to help the system administrator get started with using this power feature with Dell Compellent Storage Center and Solaris 10.

Creating Single Volume with CompCU
The example below demonstrates using CompCU to create a volume on the Storage Center SAN from the Solaris 10 host shell prompt. A 100GB volume named ceaser-compcu-testvol1 is created in the Solaris folder, and then mapped to the Solaris 10 host. From this point, the same steps would be taken on the Solaris 10 host to discover this new volume.

```
{root@ceaser} [{/root/demo}] # java -jar compcu.jar -defaultname sc9-pwdfile.cli -c "volume create -name ceaser-compcu-testvol1 -folder Solaris -server ceaser-fc -size 100g"
Compellent Command Utility (CompCU) 5.5.1.4
===============================================================================
User Name:              Admin
Host/IP Address:        sc9
Single Command:         volume create -name ceaser-compcu-testvol1 -folder Solaris -server ceaser-fc -size 100g
===============================================================================
Connecting to Storage Center: sc9 with user: Admin
Running Command: volume create -name ceaser-compcu-testvol1 -folder Solaris -server ceaser-fc -size 100g
Creating Volume using StorageType 1: storagetype='Assigned-Redundant-4096', redundancy=Redundant, pagesize=4096, diskfolder=Assigned.
Successfully mapped Volume 'ceaser-compcu-testvol1' to Server 'ceaser-fc'
Successfully created Volume 'ceaser-compcu-testvol1', mapped it to Server 'ceaser-fc' on Controller 'SN 101'

Successfully finished running Compellent Command Utility (CompCU) application.
```
Figure 13. CompCU Generated Single Volume Example

Creating a Replay with CompCU

CompCU can also be used to create a Replay of the volume created above:

```
{root@ceaser} {/root/demo} # java -jar compcu.jar -defaultname sc9-pwdfile.cli
-c "replay create -lun 200 -volume ceaser-compcu-testvol1"
Compellent Command Utility (CompCU) 5.5.1.4

===============================================================================
Connec
ting to Storage Center: sc9 with user: Admin
Running Command: replay create -lun 200 -volume ceaser-compcu-testvol1
Creating replay 'CUReplay_66022' on Volume 'ceaser-compcu-testvol1' with no expiration

Successfully finished running Compellent Command Utility (CompCU) application.
```
CompCU can be used to view the Replays that exist on the Storage Center. This can be used to grab various details about the state of volumes, replays, etc and use that information to feed into other scripts/CompCU commands to perform other related tasks. For example, the output below shows CompCU created Replay for the volume above, which lists the Index value, which can be used later.

```
{root@ceaser} /{root/demo} # java -jar compcu.jar -defaultname sc9-pwfile.cli -c "replay show"
Compellent Command Utility (CompCU) 5.5.1.4
User Name: Admin
Host/IP Address: sc9
Single Command: replay show
Connecting to Storage Center: sc9 with user: Admin
Running Command: replay show

Index  VolumeIndex  Volume       Expire  ConsistencyGroup
-------  ---------    ---------   -------  ------------------
101-111-1  111       ceaser-compcu-testvol1  Never
11/18/2011 01:05:03 pm
CUReplay_66022
101-111-3  111       ceaser-compcu-testvol1  Never
Active

Creating a View Volume of a Replay with CompCU
A View volume can then be created using the Replay created above, and then mapped to a recovery server, as exampled below:

Figure 14. Replay Listed Created from CompCU
The volume can then be imported on the recover server (fier0) to recover files or share production data from host ceaser with host fiero, etc.

```
{root@fiero} {} # format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
  0. c4t6000D3100000670000000000000006FDd0 <COMPELNT-CompellentVol-0504
     cyl 49930 alt 2 hd 8 sec 126>  SOL10OS
     /scsi_vhci/ssd@g6000d3100000670000000000000006ff
Specify disk (enter its number): ^C
```

```
{root@fiero} {} # scan.sh
Configuring the following controller port...
c0
Configuring the following controller port...
c1
```

```
{root@fiero} {} # format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
  0. c4t6000D3100000650000000000000007Cd0 <COMPELNT-CompellentVol-0504-100.00GB>
     /scsi_vhci/ssd@g6000d3100000650000000000000007cd
  1. c4t6000D3100000670000000000000006FDd0 <COMPELNT-CompellentVol-0504
     cyl 49930 alt 2 hd 8 sec 126>  SOL10OS
     /scsi_vhci/ssd@g6000d3100000670000000000000006ff
Specify disk (enter its number): ^C
```

```
{root@fiero} {} # zpool import
pool: cmldemopool
  id: 9306619001963449288
  state: ONLINE
  status: The pool was last accessed by another system.
```
action: The pool can be imported using its name or numeric identifier and the '-f' flag.
see: http://www.sun.com/msg/ZFS-8000-EY

cmdemopool ONLINE
 c4t6000D310000650000000000000007Cd0 ONLINE

{root@fiero} {} # zpool import -f cmdemopool

{root@fiero} {} # zpool list
NAME          SIZE  ALLOC   FREE    CAP  HEALTH  ALTROOT
cmdemopool  99.5G   146K  99.5G     0%  ONLINE

Rapid Deployment of Multiple Volumes with CompCU
The next example below shows the rapid deployment of several volumes and mapping them to a Solaris 10 system.

The example below is intended for demonstration purposes, but is not recommended for use in a customer environment.

{root@ceaser} {/root/demo} # for a in 0 1 2 3 4 5 6 7 8 9; do java -jar compcu.jar -defaultname sc9-pwdfile.cli -c "volume create -folder Solaris -lun 10$[a] -name ceaser-compcpu-testvol10$[a] -server ceaser-fc -size 100g"; done

Compellent Command Utility (CompCU) 5.5.1.4
===============================================================================
User Name:              Admin
Host/IP Address:        sc9
Single Command:         volume create -folder Solaris -lun 100 -name ceaser-compcpu-testvol100 -server ceaser-fc -size 100g
===============================================================================
Connecting to Storage Center: sc9 with user: Admin
Running Command: volume create -folder Solaris -lun 100 -name ceaser-compcpu-testvol100 -server ceaser-fc -size 100g
Creating Volume using StorageType 1: storagetype='Assigned-Redundant-4096', redundancy=Redundant, pagesize=4096, diskfolder=Assigned.
Successfully mapped Volume 'ceaser-compcpu-testvol100' to Server 'ceaser-fc'
Successfully created Volume 'ceaser-compcpu-testvol100', mapped it to Server 'ceaser-fc' on Controller 'SN 101'

Successfully finished running Compellent Command Utility (CompCU) application.

<SNIP>

Compellent Command Utility (CompCU) 5.5.1.4
===============================================================================
User Name:              Admin
Host/IP Address:        sc9
Single Command:         volume create -folder Solaris -lun 109 -name ceaser-compcpu-testvol109 -server ceaser-fc -size 100g
===============================================================================
Connecting to Storage Center: sc9 with user: Admin
Running Command: volume create -folder Solaris -lun 109 -name ceaser-compcpu-testvol109 -server ceaser-fc -size 100g
Creating Volume using StorageType 1: storagetype='Assigned-Redundant-4096', redundancy=Redundant, pagesize=4096, diskfolder=Assigned.

Successfully mapped Volume 'ceaser-compcpu-testvol109' to Server 'ceaser-fc'
Successfully created Volume 'ceaser-compcpu-testvol109', mapped it to Server 'ceaser-fc' on Controller 'SN 102'

Successfully finished running Compellent Command Utility (CompCU) application.

After the above is run, there are now 10 volumes created and mapped to the Solaris 10 host. The scan.sh script is then called rescan the fibre channel ports to discover the volumes. The output of format shows these new volumes are now available on the Solaris 10 host.

Figure 15. CompCU Generated Multiple Volumes Example

{root@ceaser} {/root/demo} # scan.sh
Configuring the following controller port...
c1
Configuring the following controller port...
c2
Configuring the following controller port...
c4
Configuring the following controller port...
c5

{root@ceaser} {/root/demo} # format
Searching for disks...done

c6t6000D310000065000000000000000007Ad0: configured with capacity of 100.00GB
c6t6000D310000065000000000000000007Bd0: configured with capacity of 100.00GB
c6t6000D3100000650000000000000000079d0: configured with capacity of 100.00GB
c6t6000D3100000650000000000000000072d0: configured with capacity of 100.00GB
c6t6000D3100000650000000000000000076d0: configured with capacity of 100.00GB
c6t6000D3100000650000000000000000073d0: configured with capacity of 100.00GB
c6t6000D3100000650000000000000000075d0: configured with capacity of 100.00GB
c6t6000D3100000650000000000000000078d0: configured with capacity of 100.00GB

By making this available through the Solaris 10 shell, the entire process could be scripted, including the running of the cfgadm and ZFS command(s), for example, to fully automate the provisioning of a volume and making it available to the Solaris 10 host all in one system administrator action. This
integration makes CompCU a powerful tool that can be leveraged to facilitate the automation of common the Storage Center SAN management tasks. CompCU commands can be integrated into new or existing scripts and can take advantage of the traditional automation tools built-in to the Solaris 10 operating system.

Appendix

Boot from SAN
This section will briefly cover some of the considerations around Solaris 10 Boot-from-SAN with the Dell Compellent Storage Center SAN. For more details on this process, refer to the Dell Compellent Technical Tip Boot from SAN with SPARC based Solaris Servers and Dell Compellent Storage Center, which can be found on the Compellent Knowledge Center site: http://kc.compellent.com. This guide contains a step-by-step walkthrough of the process to install a Solaris 10 system in a multipath boot-from-SAN configuration.

Beginning with Solaris 10 Update 2, boot-from-SAN is supported using the UFS file system. Note that the HBA must also have the necessary firmware release that supports boot-from-SAN as well.

Beginning with Solaris 10 Update 6, both UFS and ZFS file systems are available as root file system options for the Solaris 10 host. Either will work as expected with the Dell Compellent SAN. Refer to the UFS and ZFS sections elsewhere in this paper for specific considerations with regard to each of these file systems.

Prior to beginning the installation, verify that the Emulex or QLogic HBA used is patched to a sufficient versions of their respective drivers in order to support boot-from-SAN. Refer to the HBA’s documentation for specific details.

Either single path or multipath can be used to install the Solaris 10 OS via boot-from-SAN. If single path is used during installation, multipath can be enabled by adding the proper Compellent device entry to the /kernel/drv/scsi_vhci.conf file and activating MPxIO. Note that adding this post-installation will require a system reboot in order to activate the multipath device configuration. Then simply add the additional path and it will be added by the Solaris MPxIO software upon discovery. It is considered best practice to configure multipath whenever possible and the process is slightly simpler by having the multiple paths mapped during the OS installation.

Prior to starting the installation it is necessary to create the server object for the Solaris 10 system with the Dell Compellent Storage Center GUI. In order for the WWNs to appear, the HBA ports must be activated so that Storage Center can be made aware of them. There are a couple of options for accomplishing this, and are discussed below:

- Option 1: Use the Open Boot Prom commands show-devs, dev <device>, and .properties to display the WWN information for the HBA port(s). Refer to the Open Boot Prom section of this paper for more details. Open the Storage Center GUI and create the server objects referencing the WWNs discovered.
- Option 2: Boot the Solaris 10 system either using the Installation DVD or jumpstart server. During the boot process, the server will scan for all devices on all fibre channel
ports. When the scan is complete, go to the Storage Center GUI and create the server object using the WWNs discovered.

After creating the server object, create the volume on Storage Center, it must be mapped as LUN 0 in order for it to be properly discovered by the Solaris 10 host as a bootable device.

It is recommended to only have the boot/root OS volume mapped to the host during the boot-from-SAN installation. This will reduce confusion and remove the risk of accidentally partitioning the incorrect device which, in a worst case scenario, could cause accidental data loss.

When the Solaris 10 installation program reaches the point where it will want partition the drive, the installation program will fail with the error message:

```
One or more disks are found, but one of the following problems exists:
- Hardware failure
- Unformatted disk.
```

This error is due to the volume not yet having a proper disk label. The Solaris installer will only interact with devices that have been labeled with a Solaris disk label. The Solaris installation program will exit to a shell prompt. Use the `format` utility to label the device.

In a multipath installation there will be two devices listed. You will only need to select one of the devices and label it. Once labeled, exit out of the `format` utility and then resume the Solaris installation program by typing the command `suninstall` and then hit enter. The Solaris interactive installation program will resume.

**Open Boot Prom (OBP)**

The OBP is a low-level environment that is accessible before the Solaris 10 Operating System is running. The OBP is recognizable by the `ok` prompt that is presented for command line activity.

It is important to discuss some of the tasks that may be required to be run at the OBP level when configuring a Solaris 10 system to utilize a Dell Compellent Storage Center SAN. This section aims to provide some basic information covering some of these common tasks that may be performed in the OBP environment. For more information on the OBP and available commands refer to the resource:


**Device Name Differences Between OBP and Solaris 10**

Device names are presented quite a bit differently in the OBP environment than they are in the running Solaris 10 OS. In the OBP there is no concept of a multipath (MPxIO, for example) in the OBP, device paths are listed independently and include device class and port information in the device string.

For example, in the `fcinfo hba-port` output generated within the Solaris 10 OS below, the Emulex Fibre Channel HBA ports are represented as:
However, these same devices are presented in the OBP as shown below. Notice these also show the Compellent device (volume) that is discovered on the bus and mapped as LUN0 to the host from Storage Center:

```
{1c} ok show-devs
<SNIP>
/pci@7c0/pci@8/pci@9/emlx@0,1  → This is OS Device: /dev/cfg/c2
/pci@7c0/pci@8/pci@9/emlx@0  → This is OS Device: /dev/cfg/c1
/pci@7c0/pci@8/pci@9/emlx@0,1/fp0,0
/pci@7c0/pci@8/pci@9/emlx@0,1/fp0,0/tape
/pci@7c0/pci@8/pci@9/emlx@0,1/fp0,0/disk
/pci@7c0/pci@8/pci@9/emlx@0/fp0,0
/pci@7c0/pci@8/pci@9/emlx@0/fp0,0/tape
/pci@7c0/pci@8/pci@9/emlx@0/fp0,0/disk
<SNIP>
```

Additional OBP commands can be used to report more details on a specific device. For example, the `dev` command can be used to select a device, and then the `.properties` command can be run to get details on that particular device:

```
{1c} ok dev /pci@7c0/pci@8/pci@9/emlx@0
{1c} ok .properties
fc-boot-dev-portid       00ec0100
fc-boot-dev-portwwn      50 00 d3 10 00 00 67 09
scsi-initiator-id        00dd0100
assigned-addresses      82090010 00000000 03600000 00000000 00002000
                        82090018 00000000 03600000 00000000 00004000
                        81090020 00000000 00000000 00000000 00000100
                        82090030 00000000 03640000 00000000 00004000
port_wwn                 10 00 00 00 c9 90 bc 4e
node_wwn                 20 00 00 00 c9 90 bc 4e
<SNIP>
```

Further details on the device and connected Dell Compellent information can be achieved by using the “`<device>` select-dev” option, followed by the `show-children` command. For example, to retrieve more information on the above device, perform the following:

```
{0} ok "/pci@7c0/pci@8/pci@9/emlx@0" select-dev
{0} ok show-children
Device PortID df0200 WWPN 5000d31000006507
Device PortID df0300 WWPN 5000d31000006513
Device PortID df0400 WWPN 5000d31000006508
Device PortID df0500 WWPN 5000d31000006514
Device PortID e70400 WWPN 5000d31000ea8307
Device PortID e70401 WWPN 5000d31000ea832f
Device PortID e70500 WWPN 5000d31000ea8308
Device PortID e70501 WWPN 5000d31000ea8330
Device PortID e70600 WWPN 5000d31000ea831b
```
The above output shows all the WWNs that are discovered by this particular Emulex HBA port. In this case we can see that multiple Storage Center systems are zoned to this HBA. Of particular note is the listing that shows the “LUN 0” device that is mapped, which is the OS boot volume.

A `printenv <parameter>` can be run to get information pertaining to the designated OS boot device. For example, to list the current default boot device, run the command below. The output shows that there are two boot devices due to this system being configured in a multipath boot-from-SAN configuration. The output shows that for each of the Emulex HBA ports a LUN is assigned, which corresponds to the WWN of one of the Storage Center Fibre Channel ports in which the LUN0 device is mapped through.

```
{l1c} ok printenv boot-device
  boot-device =
  /pci@7c0/pci@0/pci@9/emlx@0/fp@0,0/disk@w5000d31000006709,0:a
  /pci@7c0/pci@0/pci@9/emlx@0,1/fp@0,0/disk@w5000d3100000670b,0:a
```

A `setenv <parameter>` can be run to change the OBP value for the given parameter. For example, if we wanted to change the default auto-boot option for the system we could run the command as below to make the change:

```
{l1c} ok printenv auto-boot?
  auto-boot? =  true

{l1c} ok setenv auto-boot? false
  auto-boot? =  false

{l1c} ok printenv auto-boot?
  auto-boot? =  false
```

To verify or confirm the Solaris 10 system is able to see the LUN0 mapped from the Dell Compellent Storage Center SAN, the `probe-scsi-all` command can be run from the OBP prompt. When the LUN0 is discovered, output will look similar to the following. Notice the LUN 0 device and COMPELNT device entries:

```
{0} ok probe-scsi-all
  /pci@7c0/pci@0/pci@9/emlx@0,1
 Device  PortID = de0400  WWPN = 5000d3100006505
 Device  PortID = de0500  WWPN = 5000d3100006511
 Device  PortID = de0600  WWPN = 5000d3100006506
 Device  PortID = de0700  WWPN = 5000d3100006512
 Device  PortID = e40400  WWPN = 5000d31000068305
 Device  PortID = e40401  WWPN = 5000d3100006832d
 Device  PortID = e40700  WWPN = 5000d31000068319
 Device  PortID = e40701  WWPN = 5000d31000068331
 Device  PortID = e40800  WWPN = 5000d3100006831a
```
Veritas
An alternative to the built-in MPxIO and UFS/ZFS file systems on Solaris 10 is the solution from Veritas. The section below is intended to provide just a brief high-level introduction to some Veritas considerations with the Dell Compellent Storage Center SAN. Please refer to the official Veritas documentation for detailed guidance on recommendations and best practices.

The Veritas DMP software provides multipath IO support. The VxFS file system provides an alternative to the native UFS and optional ZFS file systems available to Solaris 10 installations. The VxFS file system provides some advanced features not available with UFS and ZFS. Of particular note is the support for Thin Reclamation beginning with version 5.1. The following section covers various versions and the general steps necessary to install and configure Veritas DMP and VxFS with Solaris 10 SPARC using Dell Compellent Storage Center LUNs.

How to Activate the VERITAS DMP multipath on Pre Veritas 5.0 Environments
Veritas uses the udid number to determine whether the same device can be accessed via more than one path. It obtains the udid number in response to a standard SCSI INQUIRY command. Veritas inspects the VENDOR_ID and PRODUCT_ID fields and compares these against an internally held list.

Here are the rules when using the `vxddladm` command with the Storage Center:

1. The vendor id string is limited to eight characters total, so the name COMPELLENT is shortened to COMPELNT.
2. NO space between Compellent and Vol in the pid= field.
3. The two strings, COMPELNT and CompellentVol are case-sensitive and must be typed exactly as shown.
To add Compellent to the list of DMP supported JBOD arrays, run this command as root:

```
#/usr/sbin/vxddladm addjbod vid=COMPELNT pid=CompellentVol
```

List supported JBOD arrays with:

```
#vxddladm listjbod
```

![Figure 16. Veritas - Output of vxddladm Command](image)

**VERITAS DMP multipath in Veritas 5.0 Environments**

On Veritas 5.0 the Storage Center is used with a Veritas produced Array Support Library (ASL). The ASL can be obtained and installed from:

http://seer.entsupport.symantec.com/docs/322879.htm

**VERITAS DMP multipath in Veritas 5.1 Environments**

Veritas 5.1 is shipped with the Compellent Storage ASL built in to Storage Foundation.

```
# vxddladm listsupport libname=libvxcompellent.so
ATTR_NAME           ATTR_VALUE
===================================================================
LIBNAME             libvxcompellent.so
VID                 COMPELNT
PID                 Compellent Vol
ARRAY_TYPE          A/A
ARRAY_NAME          COMPELNT
```

**VERITAS DMP multipath in Veritas 6.0 Environments**

Version 6.0 also includes the Compellent Storage ASL built in to the Storage Foundation software. Therefore, it is not necessary to install any additional software outside of the Storage Foundation packages.

**Veritas Storage Foundation 6.0 with Dell Compellent Storage Center on Solaris 10 SPARC**

This section will provide an example setup and configuration of Veritas SF HA with DMP and VxFS on Solaris 10 using Dell Compellent Storage Center providing the underlying storage solution.

**Installing the Veritas 6.0 Software**

First, install the base Solaris 10 SPARC Operating System. In this example, a 64GB volume is created on the Dell Compellent Storage Center and is presented to the system called “vito”. It is recommended to boot-from-SAN whenever possible. There are distinct advantages to booting from the SAN; in particular, Storage Center Replays can be leveraged on the root OS LUN. In
the example below, the OS is installed following the recommended boot-from-SAN procedure. Refer to the Dell Compellent Boot from SAN w/SPARC base Solaris Servers Technical Tip for further details.

Solaris 10 SPARC Update 10 is installed to the boot from SAN volume and the latest 10_Recommended patch cluster is applied.

Once the OS is installed and patched with the recommended patches, the next step is to install the Veritas Storage Foundation suite, which includes the DMP and VxFS packages, among others. Refer to the Veritas documentation for details pertaining to installing Storage Foundation onto a Solaris 10 SPARC host.

1. Extract the contents of the Veritas 6.0 Software download:

   ```
   {root@vito} {/root/veritas} # gzcat
   VRTS_SF_HA_Solutions_6.0_Solaris_SPARC.tar.gz | tar xvf -
   x ./dvd1-sol_sparc/3rdpartyattributions.pdf, 657308 bytes, 1284 tape blocks
   x ./dvd1-sol_sparc/getting_started.pdf, 415782 bytes, 813 tape blocks
   x ./dvd1-sol_sparc/jumpstart_readme.txt, 8523 bytes, 17 tape blocks
   x ./dvd1-sol_sparc/readme_first.txt, 24815 bytes, 49 tape blocks
   ```

2. Change to the extracted directory and start the installer:

   ```
   {root@vito} {/root/veritas} # cd dvd1-sol_sparc/
   {root@vito} {/root/veritas/dvd1-sol_sparc} # ./installer
   ```

3. Enter “I” to select the Install a Product option:

   ![Figure 17. Veritas - Install a Product Screenshot](image)

4. Enter the appropriate option for the software to be installed. This may vary, but either option 3 or 4 will suffice for installing DMP, VxVM and VxFS packages. Type y to accept the license terms to continue with the installation.
5. Enter 3 to install all the packages and type the hostname of the system(s) in which the software will be installed. The verification process will begin.

6. Once the verification process completes, a summary of the packages to be installed will be presented. Press Enter to continue. The installation process will begin.
7. Once installation completes, the license compliance is presented. On this screen a valid license may be entered or enter 2 to use the software in evaluation mode. For the purposes of this demonstration, the evaluation mode option is selected. Then enter 1 or 2 for the Storage Foundation HA mode to utilize.
At this point the Veritas Storage Foundation installation is complete. Running the `pkginfo` command will report the VRTS related packages installed on the host for reference.

```
{root@vito} [/] # pkginfo -l | grep VRTS
PKGINST: VRTSaslapm
PKGINST: VRTSdbed
PKGINST: VRTSfsadv
PKGINST: VRTSfssdk
PKGINST: VRTScb
PKGINST: VRTSodm
PKGINST: VRTSperl
PKGINST: VRTSsfcpio60
PKGINST: VRTSsfmh
PKGINST: VRTSspt
PKGINST: VRTSsvlic
PKGINST: VRTSvxfs
PKGINST: VRTSvxvm
```

Additionally, the provided ASL library from Veritas SF includes a device definition for Compellent devices. Running `vxdisk list` will report the existing OS LUN as such:

```
{root@vito} [/] # vxdisk list
DEVICE       TYPE            DISK         GROUP        STATUS
compelnt0_0  auto:none      -            -            online invalid
```

In the next section the OS device will be encapsulated using Veritas DMP and MPIO will be enabled for the device.

**Using Veritas DMP to Multipath the root (/) UFS File System**

In this section an example use case is highlighted in which the boot-from-SAN Solaris 10 SPARC system is configured to encapsulate the OS LUN using DMP to provide MPIO support. This configuration will allow for availability of the Solaris 10 server when a single path failure occurs. This is similar to the default recommended MPxIO configuration, but will provide an example using Veritas to provide the MPIO support. After the OS LUN is encapsulated, the server will need to be rebooted to complete the configuration.
Prior to configuring DMP with the root OS LUN, the format command can be used to illustrate the multiple physical paths associated with the OS LUN. As can be seen below, the OS LUN is labeled as “SOL10OS” and is presented on two physical paths back to the fabric.

```
{root@vito} {/} # format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
  0. c0t5000D31000006909d0 <COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 336>
     /pci@1f,0/pci@1/QLGC,qlc@5/fp@0,0/ssd@w5000d31000006909,0
  1. clt5000D3100000690d0 <COMPELNT-CompellentVol-0504 cyl 49930 alt 2 hd 8 sec 336>
     /pci@1f,0/pci@1/QLGC,qlc@5,1/fp@0,0/ssd@w5000d3100000690b,0

Specify disk (enter its number): ^C
```

The following steps highlight this process on the system configured in the example installation section above. Use the `vxdiskadm` command to initiate the process to encapsulate the OS LUN.

```
{root@vito} {/} # vxdiskadm
Menu: VolumeManager/Disk

1. Add or initialize one or more disks
2. Encapsulate one or more disks
3. Remove a disk
4. Remove a disk for replacement
5. Replace a failed or removed disk
6. Mirror volumes on a disk
7. Move volumes from a disk
8. Enable access to (import) a disk group
9. Remove access to (deport) a disk group
10. Enable (online) a disk device
11. Disable (offline) a disk device
12. Mark a disk as a spare for a disk group
13. Turn off the spare flag on a disk
14. Unrelocate subdisks back to a disk
15. Exclude a disk from hot-relocation use
16. Make a disk available for hot-relocation use
17. Prevent multipathing/Suppress devices from VxVM's view
18. Allow multipathing/Unsuppress devices from VxVM's view
19. List currently suppressed/non-multipathed devices
20. Change the disk naming scheme
21. Get the newly connected/zoned disks in VxVM view
22. Change/Display the default disk layouts
list  List disk information

?  Display help about menu
?? Display help about the menuing system
q  Exit from menus

Select an operation to perform: 2

Encapsulate one or more disks
Menu: VolumeManager/Disk/Encapsulate
Use this operation to convert one or more disks to use the Volume Manager. This adds the disks to a disk group and replaces existing partitions with volumes. Disk encapsulation requires a reboot for the changes to take effect.

More than one disk or pattern may be entered at the prompt. Here are some disk selection examples:
all:        all disks
  c3 c4t2:  all disks on both controller 3 and controller 4, target 2
c3t4d2:    a single disk (in the c#t#d# naming scheme)
xyz_0:      a single disk (in the enclosure based naming scheme)
xyz_:       all disks on the enclosure whose name is xyz

Select disk devices to encapsulate:
[<pattern-list>,all,list,q,?] list

DEVICE       DISK         GROUP        STATUS
compelnt0_0  -           -            online invalid

Select disk devices to encapsulate:
[<pattern-list>,all,list,q,?] compelnt0_0

Here is the disk selected. Output format: [Device_Name]
  compelnt0_0

Continue operation? [y,n,q,?]  (default: y)  y
You can choose to add this disk to an existing disk group or to a new disk group. To create a new disk group, select a disk group name that does not yet exist.

Which disk group [<group>,list,q,?] rootdg

Create a new group named rootdg? [y,n,q,?]  (default: y)  y

Use a default disk name for the disk? [y,n,q,?]  (default: y)  y
A new disk group will be created named rootdg and the selected disks will be encapsulated and added to this disk group with default disk names.
  compelnt0_0

Continue with operation? [y,n,q,?]  (default: y)  y
The following disk has been selected for encapsulation.
Output format: [Device_Name]
  compelnt0_0

Continue with encapsulation? [y,n,q,?]  (default: y)  y
A new disk group rootdg will be created and the disk device compelnt0_0 will be encapsulated and added to the disk group with the disk name rootdg01. The compelnt0_0 disk has been configured for encapsulation. The first stage of encapsulation has completed successfully. You should now reboot your system at the earliest possible opportunity. The encapsulation will require two or three reboots which will happen automatically after the next reboot. To reboot execute the command:

```
shutdown -g0 -y -i6
```

This will update the /etc/vfstab file so that volume devices are used to mount the file systems on this disk device. You will need to update any other references such as backup scripts, databases, or manually created swap devices.

Encapsulate other disks? [y,n,q,?]  (default: n)  n

At this point it is necessary to reboot the server to activate the DMP MPIO encapsulation of the OS device.
After the host is rebooted and back online, using the `vxdisk list` command now reports the STATUS of the device indicating it is online and thin reclamation is supported as shown below:

```
{root@vito} (/) # vxdisk list
DEVICE TYPE DISK GROUP STATUS
compelnt0_0 auto:sliced rootdg01 rootdg online thinrclm
```

Executing `vxdisk list <device>` name can be used to verify the physical paths associated with the device, as highlighted in the output below:

```
{root@vito} (/) # vxdisk list compelnt0_0
Device: compelnt0_0
devicetag: compelnt0_0
type: auto
hostid: vito
disk: name=rootdg01 id=1332451418.3.vito
group: name=rootdg id=1332451419.5.vito
info: format=sliced,privoffset=1,pubslice=6,privslice=4
flags: online ready private autoconfig autoimport imported thinrclm
pubpaths: block=/dev/vx/dmp/compelnt0_0s6
char=/dev/vx/rdmp/compelnt0_0s6
privpaths: block=/dev/vx/dmp/compelnt0_0s4
char=/dev/vx/rdmp/compelnt0_0s4
guid: -
udid: COMPELNT%5FCompellent%20Vol%5F105%5F00000069-00001155
site: -
version: 2.1
iosize: min=512 (bytes) max=2048 (blocks)
public: slice=6 offset=0 len=134144640 disk_offset=67200
private: slice=4 offset=1 len=67199 disk_offset=0
update: time=1332451655 seqno=0.8
ssb: actual_seqno=0.0
headers: 0 248
configs: count=1 len=49574
logs: count=1 len=7511
Defined regions:
config priv 000017-000247{000231}: copy=01 offset=000000 enabled
config priv 000249-049591{049343}: copy=01 offset=000231 enabled
log priv 049592-057102{007511}: copy=01 offset=000000 enabled
Multipathing information:
numpaths: 2
c0t5000D31000006909d0s2 state=enabled
c1t5000D31000006909d0s2 state=enabled
```

At this point MPIO support using Veritas DMP is setup. This should be tested by failing a path (pulling cable, disabling port, etc) to verify functionality is working as expected.

**Setup VxFS LUN and use Thin Reclamation with Storage Center**

This section will cover an example setup of a Storage Center volume presented to a Solaris 10 host using the Veritas Storage Foundation provided volume manager (VxVM) and file system (VxFS) support in addition to DMP support. Beginning with Veritas 5.1, Thin Reclamation is
supported when using the VxFS (vxfs) file system on Solaris 10 SPARC. Thin Reclamation utilizes the SCSI UNMAP (also referred to in general terms as “discard”, “trim”) command to notify Storage Center when blocks have been freed from the host file system. This allows Storage Center to be made aware of the freed blocks and report these as unallocated space in the SAN. This is enabled by default and is supported with the Dell Compellent Storage Center SAN.

Use the Storage Center Management GUI or Compellent Command Utility, create a volume and map it to the Solaris 10 SPARC host. Refer to the section above titled Managing Volumes for details if needed.

Once the volume has been mapped to the Solaris 10 host and the host has been configured to discover the LUN, perform the following steps to initialize the LUN and configure it to be managed by the Veritas SF software.

1. Confirm the host can see the mapped volume on each physical. In the example below, format reports two paths to the volume, which was mapped as LUN 100:

   ```bash
   {root@vito} {/} # format
   Searching for disks...done
   c0t5000D31000006917d100: configured with capacity of 100.00GB
   c1t5000D31000006919d100: configured with capacity of 100.00GB
   AVAILABLE DISK SELECTIONS:
   0. c0t5000D31000006909d0 <COMPELNT-CompellentVol-0504 cyl
   49930 alt 2 hd 8 sec 336> SOL10OS
       /pci@1f,0/pci@1/QLGC,qlc@5/fp@0,0/ssd@w5000d31000006909,0
   1. c0t5000D31000006917d100 <COMPELNT-CompellentVol-0505 cyl
   49930 alt 2 hd 8 sec 525>
       /pci@1f,0/pci@1/QLGC,qlc@5/fp@0,0/ssd@w5000d31000006917,64
   2. c1t5000D3100000690Bd0 <COMPELNT-CompellentVol-0504 cyl
   49930 alt 2 hd 8 sec 336> SOL10OS
       /pci@1f,0/pci@1/QLGC,qlc@5,1/fp@0,0/ssd@w5000d3100000690b,0
   3. c1t5000D31000006919d100 <COMPELNT-CompellentVol-0505 cyl
   49930 alt 2 hd 8 sec 525>
   /pci@1f,0/pci@1/QLGC,qlc@5,1/fp@0,0/ssd@w5000d31000006919,64
   Specify disk (enter its number):
   
   2. Next, run the `vxdisk scandisks` command to initiate the discovery of the LUN at the Veritas layer. Then run `vxdisk list` to see the newly discovered device and verify MPIO is enabled.

   ```bash
   {root@vito} {/} # vxdisk scandisks
   {root@vito} {/} # vxdisk list
   DEVICE TYPE DISK GROUP STATUS
   compelnt0_0 auto:sliced rootdg01 rootdg online thinrclm
   compelnt0_1 auto - - nolabel
   {root@vito} {/} # vxdisk list compelnt0_1
   Device: compelnt0_1
devicetag: compelnt0_1
type: auto
flags: nolabel private autoconfig
pubpaths: block=/dev/vx/dmp/compelnt0_1
char=/dev/vx/rdmp/compelnt0_1
guid: -
Dell Compellent Storage Center Solaris Best Practices

3. The next step is to initialize the device. The vxassist CLI interface can be used as well as running commands directly from the CLI to the device. Refer to the Veritas documentation for further details.

   {root@vito} {} # /etc/vx/bin/vxdisksetup -i compelnt0_1
   format=cdsdisk privlen=65536

   {root@vito} {} # vxdisk list
   DEVICE   TYPE        DISK         GROUP        STATUS
   compelnt0_0 auto:sliced rootdg01 rootdg online thinrclm
   compelnt0_1 auto:cdsdisk - - online thinrclm

4. Create the disk group(s):

   {root@vito} {} # vxdg init testvol1dg testvol01=compelnt0_1

5. Create the sub-disk(s):

   {root@vito} {} # vxdg list compelnt0_1 | grep public
   public: slice=2 offset=65792 len=209640208 disk_offset=0

   {root@vito} {} # vxmake -g testvol1dg sd testvol1sd
   testvol01,0,209640208

6. Create the plex:

   {root@vito} {} # vxmake -g testvol1dg plex testvol1p sd=testvol1sd

7. Create and activate the volume:

   {root@vito} {} # vxmake -g testvol1dg -U fsgen vol testvol1v
   plex=testvol1p

   {root@vito} {} # vxvol -g testvol1dg start testvol1v

   {root@vito} {} # vxdisk list
   DEVICE   TYPE        DISK         GROUP        STATUS
   compelnt0_0 auto:sliced rootdg01 rootdg online thinrclm
   compelnt0_1 auto:cdsdisk testvol01 testvol1dg online thinrclm

8. Create the VxFS file system on the volume and mount the device to a mount point on the host.

   {root@vito} {} # mkfs -F vxfs -o largefiles -o bsize=4096
   /dev/vx/rdsk/testvol1dg/testvol1v
   version 9 layout
   209640208 sectors, 26205026 blocks of size 4096, log size 16384
   blocks
   rcq size 1024 blocks
   largefiles supported
The above example is just one file system setup possibility. Veritas documentation should be consulted for further information, specifically pertaining to features such as largefiles support and performance tuning options.

To demonstrate the support for Thin Reclamation, the example below details a situation in which a thinly provisioned volume is populated with data to a point where 82% of the reported volume capacity is utilized. In a traditional Thin Provisioned implementation, as the data is removed at the OS and file system layer, the OS reports the newly free space; however, the SAN continues to report that usage of the blocks equaling the 82% level.

By employing the “thinrclm” support included with Veritas Storage Foundation, the SAN is able to accurately report the available capacity due to the information about those freed blocks being transmitted to the SAN via the SCSI UNMAP command.

Figures 23 and 24 below show the reported usage of the 100GB file system created above. These show the device is approximately 82% utilized.

Figure 23. Veritas - Host Reported Utilization - Pre File Deletes
After removing approximately 36GB of data, the file system reports a usage of 47%, whereas the Storage Center GUI reports approximately 81GB used, which is approximately 82% of the total volume size.

```
[root@vito /testvol1]# cd
[root@vito /]# df -h /testvol1/
Filesystem size used avail capacity mounted on
/dev/vx/dsk/testvol1d/testvol1 100G 81G 19G 82% /testvol1
[root@vito /]# cd /testvol1/
[root@vito /testvol1]# du -sh *
12G iso0
12G iso1
12G iso2
12G iso3
12G iso4
12G iso5
12G iso6
Ok lost-found
[root@vito /testvol1]# rm -rf iso0 iso1 iso2 iso3 iso4 iso5 iso6
[root@vito /testvol1]# df -h /testvol1/
Filesystem size used avail capacity Mounted on
/dev/vx/dsk/testvol1d/testvol1 100G 72G 27G 78% /testvol1
[root@vito /testvol1]# df -h /testvol1/
Filesystem size used avail capacity Mounted on
/dev/vx/dsk/testvol1d/testvol1 100G 49G 51G 49% /testvol1
[root@vito /testvol1]#
```

Figure 25. Veritas - Host Reported Utilization - Post File Deletes
The `fsadm` command can be run to clear the free blocks associated with the file system deletes performed above. This command can be found in the `/opt/VRTS/bin` directory. Running the command against the vxfs file system mount point, as shown in Figure 27, will result in the SCSI UNMAP command being passed to the Storage Center SAN, thus resulting in the SAN being made aware of the freed blocks, and reporting as shown in Figure 28 below.
After running the `fsadm` command, refresh the Statistics pane in the Storage Center GUI for the VxFS volume. The actual used blocks will now be reported showing the Storage Center SAN is now made aware of the freed blocks from the host.

![Storage Center GUI](image)

**Figure 28. Veritas - Storage Center Statistics for Volume After fsadm**

If there are active Replays that are pointing to the blocks deleted, the blocks will remain frozen and not unmapped by the SAN until the Replay expires.

---

**Expand a Storage Center Volume using a VxFS File System**

A common management task a system or storage administrator will often need to execute is the need to expand the available capacity of a file system. The Storage Center Volume Expansion feature provides the ability to expand a volume to meet new capacity and utilization requirements. The Veritas VxFS file system supports the ability to resize without taking the file system offline.

This section will highlight an example scenario in which a Solaris 10 SPARC host using a VxFS file system on a Storage Center volume is nearing 100% utilization. Steps will be highlighted showing one method for expanding the underlying Storage Center volume and the VxFS file system without impacting the availability of the data or halting I/O. The volume will be expanded from 100GB to 200GB.

The steps below will involve the low-level reconfiguration of the file system; as such there is always the risk of data loss. Therefore, it is considered best practice to take a replay of the volume prior to executing any of the steps below.

First, using the Storage Center GUI or CompCU, take a replay of the volume to be expanded. The steps below will use the Storage Center GUI to execute the steps necessary to expand the
volume on the SAN, and then use the CLI on the Solaris 10 host to perform the Veritas commands necessary to expand the file system.

Expand the volume by executing the following steps within the Storage Center GUI:

1. Login to the Dell Compellent Storage Center Management GUI as a user with Administrative privileges.
2. Navigate to the volume to be expanded.
3. Right-click on the volume and select Expand Volume.
4. Enter the new size of the volume. Click **Expand Volume Now**.
5. Confirm the new size by clicking on the General tab for the volume.

![Dell Compellent Storage Center Management GUI](image)

**Figure 29. Veritas - Storage Center Volume Size after Expanded**

Expand the file system using the native OS and Veritas management commands via the CLI:

1. Login to the Solaris 10 host, and then become the root user.
2. Using the `df` command, it is seen the file system still reports the original size of 100GB:

   ```
   {root@vito} [/testvol1] # df -h .
   Filesystem    size  used  avail  capacity Mounted on
   /dev/vx/dsk/testvol1dg/testvol1v 100G  93G  7.2G   93%  /testvol1
   ```

3. Use the format command to resize the disk geometry. Note the original partition layout of the device so as to replicate this same layout prior to labeling the device with the new disk geometry:

   ```
   {root@vito} [/testvol1] # format
   Searching for disks...done
   AVAILABLE DISK SELECTIONS:
   ```
0. c0t5000D31000006909d0 <COMPELNT-CompellentVol-0504 cyl 49930
alt 2 hd 8 sec 336> SOL100S
pci@1f,0/pci@1/QLGC,qlc@5/fp@0,0/ssd@w5000d31000006909,0
1. c0t5000D31000006917d100 <COMPELNT-CompellentVol-0505 cyl 49930
alt 2 hd 8 sec 525>
pci@1f,0/pci@1/QLGC,qlc@5/fp@0,0/ssd@w5000d31000006917,64
2. c1t5000D3100000690Bd0 <COMPELNT-CompellentVol-0504 cyl 49930
alt 2 hd 8 sec 336> SOL100S
pci@1f,0/pci@1/QLGC,qlc@5,1/fp@0,0/ssd@w5000d3100000690B,0
3. c1t5000D31000006919d100 <COMPELNT-CompellentVol-0505 cyl 49930
alt 2 hd 8 sec 525>
pci@1f,0/pci@1/QLGC,qlc@5,1/fp@0,0/ssd@w5000d31000006919,64
Specify disk (enter its number): 1
selecting c0t5000D31000006917d100
[disk formatted]

FORMAT MENU:
disk - select a disk
type - select (define) a disk type
partition - select (define) a partition table
current - describe the current disk
format - format and analyze the disk
repair - repair a defective sector
label - write label to the disk
analyze - surface analysis
defect - defect list management
backup - search for backup labels
verify - read and display labels
save - save new disk/partition definitions
inquiry - show vendor, product and revision
volname - set 8-character volume name
!<cmd> - execute <cmd>, then return
quit
format> type

AVAILABLE DRIVE TYPES:
0. Auto configure
1. Quantum ProDrive 80S
2. Quantum ProDrive 105S
3. CDC Wren IV 94171-344
4. SUN0104
5. SUN0207
6. SUN0327
7. SUN0340
8. SUN0424
9. SUN0535
10. SUN0669
11. SUN1.0G
12. SUN1.05
13. SUN1.3G
14. SUN2.1G
15. SUN2.9G
16. Zip 100
17. Zip 250
18. Peerless 10GB
19. COMPELNT-CompellentVol-0504
20. COMPELNT-CompellentVol-0505
21. other
Specify disk type (enter its number) [20]: 0
c0t5000D31000006917d100: configured with capacity of 199.99GB
<COMPELNT-CompellentVol-0505 cyl 63935 alt 2 hd 32 sec 205>
selecting c0t5000D31000006917d100
[disk formatted]
PARTITION MENU:
0 - change '0' partition
1 - change '1' partition
2 - change '2' partition
3 - change '3' partition
4 - change '4' partition
5 - change '5' partition
6 - change '6' partition
7 - change '7' partition
select - select a predefined table
modify - modify a predefined partition table
name - name the current table
print - display the current table
label - write partition map and label to the disk
!<cmd> - execute <cmd>, then return
quit

partition> pr
Current partition table (default):
Total disk cylinders available: 63935 + 2 (reserved cylinders)

<table>
<thead>
<tr>
<th>Part</th>
<th>Tag</th>
<th>Flag</th>
<th>Cylinders</th>
<th>Size</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>root</td>
<td>wm</td>
<td>0 - 39</td>
<td>128.12MB</td>
<td>(40/0/0)</td>
</tr>
<tr>
<td>1</td>
<td>swap</td>
<td>wu</td>
<td>40 - 79</td>
<td>128.12MB</td>
<td>(40/0/0)</td>
</tr>
<tr>
<td>2</td>
<td>backup</td>
<td>wu</td>
<td>0 - 63934</td>
<td>199.99GB</td>
<td>(63935/0/0)</td>
</tr>
<tr>
<td>3</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0)</td>
</tr>
<tr>
<td>4</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0)</td>
</tr>
<tr>
<td>5</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0)</td>
</tr>
<tr>
<td>6</td>
<td>usr</td>
<td>wm</td>
<td>80 - 63934</td>
<td>199.74GB</td>
<td>(63855/0/0)</td>
</tr>
<tr>
<td>7</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0)</td>
</tr>
</tbody>
</table>

partition> 0
Enter partition id tag[root]: unassigned
Enter partition permission flags[wm]:
Enter new starting cyl[0]: 0
Enter partition size[262400b, 40c, 39e, 128.12mb, 0.13gb]: c

partition> 1
Enter partition id tag[swap]: unassigned
Enter partition permission flags[wu]:
Enter new starting cyl[40]: 0
Enter partition size[262400b, 40c, 39e, 128.12mb, 0.13gb]: 0c

partition> 6
Enter partition id tag[usr]: unassigned
Enter partition permission flags[wm]:
Enter new starting cyl[80]: 0
Enter partition size[418888800b, 63855c, 63854e, 204535.55mb, 199.74gb]: 0c

partition> y
'y' is not expected.

partition> 7
Enter partition id tag[unassigned]:
Enter partition permission flags[wm]:
Enter new starting cyl[0]:
Enter partition size[419407040b, 63934c, 63933e, 204788.59mb, 199.99gb]:

```
partition> pr
```
Current partition table (unnamed):
Total disk cylinders available: 63935 + 2 (reserved cylinders)

```
partition> lab
Ready to label disk, continue? yes
```

4. Next, use the vxdisk scandisks command to rescan the disk device, letting Veritas recognize the new physical size of the disk:

```
{root@vito} (/testvol1) # vxdisk scandisks
```

5. Attempting to expand the file system at this point would lead to the following error:

```
{root@vito} (/testvol1) # vxdisk -g testvol01dg resize testvol01
testvol01dg: resize failed: Cannot remove last disk in disk group
```

6. Create a temporary volume to hold the disk group VxVM information during the resize process. In this case a 1GB volume was created and mapped to the host. The host was configured to rescan the HBA ports using `cfgadm` and the newly mapped volume was labeled using the `format` command.
7. The temporary volume was then initialized into the VxVM configuration and added to the existing disk group to allow the volume expansion to complete. The following Veritas commands are executed to complete the expansion of the source volume:

```
{root@vito} {/} # vxdisk scandisks

{root@vito} {/} # vxdisk list
DEVICE TYPE DISK GROUP STATUS
compelnt0_0 auto:sliced rootdg01 rootdg online thinrclm
compelnt0_1 auto:cdsdisk testvol01 testvol1dg online thinrclm
compelnt0_2 auto:none - - online invalid

{root@vito} {/} # /etc/vx/bin/vxdisksetup -i compelnt0_2 format=cdsdisk privlen=65536

{root@vito} {/} # vxdg -g testvol1dg adddisk testvol02=compelnt0_2

{root@vito} {/} # vxdisk list
DEVICE TYPE DISK GROUP STATUS
compelnt0_0 auto:sliced rootdg01 rootdg online thinrclm
compelnt0_1 auto:cdsdisk testvol01 testvol1dg online thinrclm
compelnt0_2 auto:cdsdisk testvol02 testvol1dg online thinrclm

{root@vito} {/} # vxdisk -g testvol1dg resize testvol01

{root@vito} {/} # /etc/vx/bin/vxresize -g testvol1dg testvol1v +99g

{root@vito} {/} # df -h /testvol1/
Filesystem size used avail capacity Mounted on
/dev/vx/dsk/testvol1dg/testvol1v 199G 93G 105G 47% /testvol1

8. The temporary volume created in step 6 above can now be removed. The steps below highlight the process to accomplish this stage:

```
{root@vito} {/testvol1} # vxdisk list
DEVICE TYPE DISK GROUP STATUS
compelnt0_0 auto:sliced rootdg01 rootdg online thinrclm
compelnt0_1 auto:cdsdisk testvol01 testvol1dg online thinrclm
compelnt0_2 auto:cdsdisk testvol02 testvol1dg online thinrclm

{root@vito} {/testvol1} # vxdg -g testvol1dg rmdisk testvol02

{root@vito} {/testvol1} # vxdisk list
DEVICE TYPE DISK GROUP STATUS
compelnt0_0 auto:sliced rootdg01 rootdg online thinrclm
compelnt0_1 auto:cdsdisk testvol01 testvol1dg online thinrclm
compelnt0_2 auto:cdsdisk - - online thinrclm

{root@vito} {/testvol1} # vxdisk rm compelnt0_2

9. Execute the native Solaris 10 OS commands to remove the device from the system as outlined below:

a. Offline the LUNs:

```
{root@vito} {/testvol1} # luxadm -e offline /dev/rdsk/c0t5000d31000006909d150s2

{root@vito} {/testvol1} # luxadm -e offline /dev/rdsk/c1t5000d31000006908d150s2
```
b. Login to the Storage Center GUI and remove the mappings for the 1GB temporary volume.

c. Run the `cfgadm -o force_update -c configure <port>` command to rescan the HBA ports.

d. Run the `devfsadm -C` command to remove the device entries.

10. Finally, clean up the Veritas device information to no longer have reference to the temporary volume. These steps are exampled below:

```bash
{root@vito} {/testvol1} # vxddladm stop eventsource
{root@vito} {/testvol1} # mv /etc/vx/disk.info /etc/vx/disk.info.old
{root@vito} {/testvol1} # mv /etc/vx/array.info /etc/vx/array.info.old
{root@vito} {/testvol1} # vxconfigd -k -x cleartempdir
{root@vito} {/testvol1} # vxddladm start eventsource
{root@vito} {/testvol1} # vxdisk list
```

```
DEVICE       TYPE            DISK         GROUP        STATUS
compelnt0_0  auto:sliced     rootdg01     ro            otdg        online thinrcilm
compelnt0_1  auto:cdsdisk    testvol01    testvol1dg   online thinrcilm
```

It is important to reiterate the steps above for volume expansion are just one method by which expansion can be accomplished. As with other UNIX related tasks there is often more than one way to accomplish the same task. For example, the above could be accomplished also by simply creating a new volume of the desired size to increase the VxFS file system size by, and add that volume to the existing volume group. Such an approach would have the added benefit of balancing volumes across multiple Storage Center controllers, which could lead to increased performance of the host file system. Careful consideration should be taken to use the method that is most appropriate for the customer environment.