Read Me First

Avoiding a Potential Data Integrity Problem

This problem may affect all users of RAID Array 450, HSZ50, HSJ50 and HSZ40. If you are running HSOF V3.0 or V5.0 software in configurations containing mirrorsets (RAID 0) or striped mirrorsets (RAID 0+1), you are exposed to a potential data integrity problem.

To avoid this problem, all users must implement one of the following two procedures. In addition, DIGITAL UNIX users must complete an additional procedure to prevent the host from attempting to override the maximum transfer size (see the procedures for either DIGITAL UNIX V3.2 or DIGITAL UNIX 4.0, below).

1. If you are installing StorageWorks Command Console (SWCC) with your initial configuration, you do not have to do anything additional. SWCC's default settings correctly set the MAXIMUM_CACHED_TRANSFER_SIZE parameter to 1024.

   As you use SWCC, do not change the MAXIMUM_CACHED_TRANSFER_SIZE parameter.

2. If you are using the Command Line Interpreter (CLI) and not using SWCC, you must perform the following procedure as part of your initial setup, (that is, incorporate the procedure below as part of your configuration process, and prior to loading any data onto the RAID Array). Note that the procedure must be carried out on each mirrorset or striped mirrorset in the subsystem.

   Procedure

   To access the RAID Controller and use the Command Line Interpreter (CLI), you must make a serial connection between your maintenance terminal, PC, or tip connection, and the controller.

   1. Issue the following CLI command:

      ```
      CLI> SET Dxyy MAXIMUM_CACHED_TRANSFER_SIZE = 1024
      ```

      where **x** equals the Target ID of the mirrorset and **yy** equals the LUN ID of the mirrorset.

   2. Ensure that the mirrorset uses read cache by issuing the following CLI command:

      ```
      CLI> SHOW Dxyy
      ```

      Look for READ CACHE ENABLED in the display. If read cache is disabled, enable it by issuing the following CLI command:

      ```
      CLI> SET Dxyy READ_CACHE
      ```

   3. Repeat Step 1 and 2 for all mirrorsets and striped mirrorsets

   4. Restart both controllers by issuing the following CLI commands.

      ```
      CLI> RESTART OTHER_CONTROLLER
      CLI> RESTART THIS_CONTROLLER
      ```

   5. Remount the affected filesystems or devices and restart your applications as appropriate to your host environment.

Verification

Use the SHOW Dxyy command to verify that the MAXIMUM_CACHED_TRANSFER_SIZE parameter is set to 1024 and that READ CACHE is enabled for all mirrorsets and striped mirrorsets.

DIGITAL UNIX SPECIFIC PROCEDURE

In addition to the above, DIGITAL UNIX users must also do the following:

Problem

All versions of DIGITAL UNIX have the ability to permit applications software to request 16 megabyte raw I/O reads and writes. So even with the controller's maximum cache transfer size set to 512k bytes (1024 blocks) an application using raw I/O may set up the right conditions for the HSOF V3.0/5.0 mirror set problem to occur.

Resolution

DIGITAL's recommended solution is to tune DIGITAL UNIX so that it will not perform reads and writes larger than the 512K byte maximum cache transfer size. Using the following instructions, DIGITAL UNIX can be tuned so that it will, instead, break up such requests into multiple 512K byte requests.

This tuning can be performed manually or with the aid of specially created scripts. To ease the process and reduce the possibility of human error, DIGITAL recommends using the scripts. A C language program is also available to verify the results of the tuning.

These two scripts (v3_512k_max_rec.csh and v4_512k_max_rec.csh) and the C language program v3_DumpTable.c can be obtained from DIGITAL's Customer Support Centers.

I. TO MANUALLY TUNE DIGITAL UNIX VERSIONS PRIOR TO V4.0 FOR A 512K BYTE MAXIMUM TRANSFER SIZE -

A. Become root.

B. **`cd /usr/sys/data`**

C. **`cp cam_data.c cam_data.c.orig /* save original file */`**

D. In the file cam_data.c, using a text editor, search for the HSZ20, HSZ40, and HSZ50 entries.

E. Change the DEC_MAX_REC field in each entry to (1024 * 512) bytes using a text editor.

HSZ20 Example:

****************************** Change From *********************************

```
/* HSZ20 */
{"DEC     HSZ2", 12, DEV_HSZ20, (ALL_DTYPE_DIRECT << DTYPE_SHFT) |
SZ_HARD_DISK
  | SZ_RAID ,
  (struct pt_info *)ccmn_hsx00_sizes, NULL, DEC_MAX_REC, NO_DENS_TAB,
  NO_MODE_TAB, (SZ_DYNAMIC_GEOM | SZ_DISPERSE_QUE | SZ_REORDER),
  NO_OPT_CMDS, 90, 74, DD_REQSNS_VAL |
  DD_INQ_VAL, 36, 160
},
```

****************************** Change To ************************************

```
/* HSZ20 */
{"DEC     HSZ2", 12, DEV_HSZ20, (ALL_DTYPE_DIRECT << DTYPE_SHFT) |
SZ_HARD_DISK
  | SZ_RAID ,
  (struct pt_info *)ccmn_hsx00_sizes, NULL, (1024 * 512), NO_DENS_TAB,
  NO_MODE_TAB, (SZ_DYNAMIC_GEOM | SZ_DISPERSE_QUE | SZ_REORDER),
  NO_OPT_CMDS, 90, 74, DD_REQSNS_VAL |
  DD_INQ_VAL, 36, 160
},
```

HSZ40 Example:

******************************** Change From ********************************

```
/* HSZ40 */
{"DEC     HSZ4", 12, DEV_HSZ40, (ALL_DTYPE_DIRECT << DTYPE_SHFT) |
SZ_HARD_DISK
  | SZ_RAID ,
  (struct pt_info *)ccmn_hsx01_sizes, NULL, DEC_MAX_REC, NO_DENS_TAB,
  NO_MODE_TAB,
  (SZ_LONG_STO_RETRY | SZ_DYNAMIC_GEOM | SZ_DISPERSE_QUE | SZ_REORDER),
  NO_OPT_CMDS, 90, 74, DD_REQSNS_VAL |
  DD_INQ_VAL,
  36, 160/* HSZ40 */
```

****************************** Change To ************************************

```
{"DEC     HSZ4", 12, DEV_HSZ40, (ALL_DTYPE_DIRECT << DTYPE_SHFT) |
SZ_HARD_DISK
  | SZ_RAID ,
  (struct pt_info *)ccmn_hsx01_sizes, NULL, (1024 * 512), NO_DENS_TAB,
  NO_MODE_TAB,
  (SZ_LONG_STO_RETRY | SZ_DYNAMIC_GEOM | SZ_DISPERSE_QUE | SZ_REORDER),
  NO_OPT_CMDS, 90, 74, DD_REQSNS_VAL |
  DD_INQ_VAL,
  36, 160
},
```

HSZ50 Example:

******************************** Change From ******************************

```
/* HSZ5x */
{"DEC     HSZ5", 12, DEV_HSZ50, (ALL_DTYPE_DIRECT << DTYPE_SHFT) |
SZ_HARD_DISK
  | SZ_RAID ,
  (struct pt_info *)ccmn_hsx01_sizes, NULL, DEC_MAX_REC, NO_DENS_TAB,
  NO_MODE_TAB,
  (SZ_LONG_STO_RETRY | SZ_DYNAMIC_GEOM | SZ_DISPERSE_QUE | SZ_REORDER),
  NO_OPT_CMDS, 90, 74, DD_REQSNS_VAL |
  DD_INQ_VAL,
  36, 160
},
```

******************************** Change To *********************************

```
/* HSZ5x */
{"DEC     HSZ5", 12, DEV_HSZ50, (ALL_DTYPE_DIRECT << DTYPE_SHFT) |
SZ_HARD_DISK
  | SZ_RAID ,
  (struct pt_info *)ccmn_hsx01_sizes, NULL,(1024 * 512), NO_DENS_TAB,
  NO_MODE_TAB,
  (SZ_LONG_STO_RETRY | SZ_DYNAMIC_GEOM | SZ_DISPERSE_QUE | SZ_REORDER),
  NO_OPT_CMDS, 90, 74, DD_REQSNS_VAL |
  DD_INQ_VAL,
  36, 160
},
```

F. Write the file and leave the editor.

G. Rebuild the kernel.

H. Reboot the system with the kernel just created. [NOTE: The maximum record size change will not become effective until the system is rebooted.]

II. TO TUNE VIA SCRIPT DIGITAL UNIX VERSIONS PRIOR TO V4.0 FOR A 512K BYTE MAXIMUM TRANSFER SIZE -

A. Obtain the script v3_512k_max_rec.csh from a DIGITAL Customer Support Center:

B. Place the script file in directory

```
/usr/tmp
```

C. Become root.

D. Execute the script v3_512k_max_rec.csh. It will perform steps I.B through I.F above.

E. Rebuild the kernel.

G. Reboot the system with the kernel just created. [NOTE: The maximum record size change will not become effective until the system is rebooted.]

III. TO VERIFY THE CURRENT VALUE OF THE MAXIMUM TRANSFER SIZE PARAMETER ON DIGITAL UNIX SYSTEMS PRIOR TO V4.0 -

A. Obtain the file v3_DumpTable.c from a DIGITAL Customer Support Center.

B. Place the file in directory:

```
/usr/tmp
```

C. Compile file DumpTable.c

D. Execute the resulting program. It will print out an internal DIGITAL UNIX kernel table which will show the value as in the following "before and after tuning" examples.

Before tuning, the HSZ entries would appear as:

```
DEC     HSZ2  16777216
DEC     HSZ4  16777216
DEC     HSZ5  16777216
```

After tuning, they would appear as:

```
DEC     HSZ2  524288
DEC     HSZ4  524288
DEC     HSZ5  524288
```

IV. TO MANUALLY TUNE DIGITAL UNIX VERSIONS V4.0 AND LATER FOR a 512K BYTE MAXIMUM TRANSFER SIZE -

A. Become root.

B. **`cd /etc`**

C. **`cp ddr.dbase ddr.dbase.orig     /* save original file */`**

D. In the file ddr.dbase, using a text editor, search for "HSZx"
Before modification, the entry will look as follows:

```
SCSIDEVICE
    #
    # Matches everyone in the HSZx family
    #
    Type = disk
    Name = "DEC" "HSZ"
    #
    PARAMETERS:
        TypeSubClass        = hard_disk, raid
        BlockSize           = 0
        BadBlockRecovery    = disabled
        DynamicGeometry     = true
        DisperseQueue       = true
        LongTimeoutRetry    = enabled
        ReadyTimeSeconds    = 90
        TagQueueDepth       = 74
        RequestSenseLength  = 160
        PwrMgmt_Capable     = false
```

E. Add the MaxTransferSize parameter with a value assigned of 0x80000.
   After modification, the entry should look as follows:

```
SCSIDEVICE
    #
    # Matches everyone in the HSZx family
    #
    Type = disk
    Name = "DEC" "HSZ"
    #
    PARAMETERS:
        TypeSubClass        = hard_disk, raid
        BlockSize           = 0
        BadBlockRecovery    = disabled
        DynamicGeometry     = true
        DisperseQueue       = true
        LongTimeoutRetry    = enabled
        ReadyTimeSeconds    = 90
        TagQueueDepth       = 74
        RequestSenseLength  = 160
        PwrMgmt_Capable     = false
        MaxTransferSize     = 0x80000
```

F. Write the file and leave the editor.

G. Rebuild the database using the following command:

```
ddr_config -c
```

H. Reboot the system. [NOTE: The maximum record size change will not become effective until the system is rebooted.]

V. TO TUNE VIA SCRIPT DIGITAL UNIX VERSIONS V4.0 AND LATER FOR A 512K BYTE MAXIMUM TRANSFER SIZE -

A. Obtain the script v4_512k_max_rec.csh from a DIGITAL CustomerSupport Center.

B. Place the file in directory

```
/usr/tmp
```

C. Become root.

D. Execute the script v4_512k_max_rec.csh. It will perform steps IV.B through IV.G above.

E. Reboot the system. [NOTE: The maximum record size change will not become effective until the system is rebooted.]

VI. TO RETUNE DIGITAL UNIX VERSIONS PRIOR TO V4.0 FOR THE ORIGINAL, I.E. 16 MEGABYTE, MAXIMUM RAW I/O TRANSFER SIZE -

A. Become root.

B. **`cd /usr/sys/data`**

C. **`cp -p cam_data.c.orig cam_data.c.      /* restore original file */`**

D. Rebuild the kernel.

E. Reboot the system. [NOTE: The maximum record size change will not become effective until the system is completed.]

VII. TO RETUNE DIGITAL UNIX VERSIONS V4.0 AND LATER FOR THE ORIGINAL, I.E. 16 MEGABYTE, MAXIMUM RAW I/O TRANSFER SIZE -

A. Become root.

B. **`cd /etc`**

C. **`cp -p ddr.dbase.orig ddr.dbase        /* restore original file */`**

D. Rebuild the database using the following command:

```
ddr_config -c
```

E. Reboot the system. [NOTE: The maximum record size change will not become effective until the system is rebooted.]

VIII. TO VERIFY THE CURRENT VALUE OF THE MAXIMUM TRANSFER SIZE PARAMETER FOR DIGITAL UNIX VERSIONS V4.0 AND LATER, EXECUTE THE FOLLOWING COMMAND:

```
ddr_config -s 0 DEC HSZ
```

When a 512k maximum transfer size has been specified, there will be one line, among many, of the output of this command that looks:

```
MaxTransferSize = 0x80000
```

When a 16 megabyte transfer size has been specified, there will be one line, among many, of the output of this command that looks like:

```
MaxTransferSize = 0x1000000
```

ADDITIONAL COMMENTS:

There may be a slight performance degradation experienced as a result of tuning for a 512K byte maximum transfer size.

The above workarounds have no effect on problems that may already have occurred. They serve only to prevent applications software from creating the conditions which may lead to the HSOF V3.0/5.0 mirrorset problem.

ADDITIONAL INFORMATION:

A patch that will correct this problem in HSOF V3.0 and V5.0 for all host environments has been identified and is currently being tested. Patch release is expected within two weeks. All MDDS contract Customers and Customers who have registered their controllers will receive a letter containing the patch as soon as it is released.