****

README.PDF

****

This file presents general information about the LSI Fusion-MPT (TM) (Message Passing Technology) device drivers for Linux(TM). It also describes the features and use of the device drivers for the Linux operating system environment.

This file is divided into the following sections:

******************************* Introduction **

The mpt2sas drivers are free software and are supported in source form. These drivers are distributed in the hope that they will be useful, but without any warranty and without even the implied warranty of merchantability or fitness for a particular purpose. You can redistribute them and/or modify them under the terms of version 2 or later of the GNU Public License as published by the Free Software Foundation. You should have received a copy of this license with your Linux kernel source tree (/usr/src/linux/COPYING). For detailed information on the GNU Public License, contact the Free Software Foundation, Inc at 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 or at URL http://www.gnu.org/copyleft/gpl.html.

=== Features ===

o SAS/SATA flexibility, supporting 1.5 Gb/s, 3.0 Gb/s, and 6 Gb/s devices
o PCI Express host interfaces
o Supporting MSI-X Interrupt Routing (RHEL5, SLES10, SLES11)
o Fusion-MPT Architecture and common software interface
o Embedded CPU in ASIC performs RAID operations
o Integrated RAID available on certain boards
o Supported RAID levels are (0, 1, 1E, 10)

=== LSI Devices Supported ===

o SAS2004, SAS2008, SAS2108, and SAS2116

***************** Fusion-MPT Linux Drivers **

The mpt2sas drivers are provided in binary and source form to provide the greatest flexibility to LSI customers. The binaries are suitably formatted for use as installation diskettes or post-install binary upgrades. The source may be added to an existing kernel for custom kernel builds.

NOTE: THERE ARE TWO DRIVERS!!
mpt2sas.ko is for Red Hat (RHEL5), and SuSE (SLES10/SLES11)
mpt2sasbtm.ko is for Red Hat (RHEL4), and SuSE (SLES9)

=== Installing to a Fusion-MPT Controller ===

LSI provides 1.44MB images that are suitably formatted to use as driver update disks for those installations where there is no bundled driver or when it's necessary to complete the install with the latest driver:

    Red Hat RHEL4 (i686, x86_64)
    Red Hat RHEL5 (i686, ia64, x86_64, ppc)
    SuSE SLES10, SLES11 (i586, ia64, x86_64, ppc)

Here are some examples creating a driver update disk. In this example the driver version is (01.255.06.00-1). Here is the procedure:

```
    Extract the packaging from Linux system:
    # tar -zxvf mpt2sas-release.tar.gz
    # cd disks-1

    for RHEL5 Update 3, architecture = x86_64
    # gunzip mpt2sas-01.255.06.00-1-rhel5.3.x86_64.dd.gz
    # dd if=mpt2sas-01.255.06.00-1-rhel5.3.x86_64.dd of=/dev/fd0

    for SLES10 Service Pack 2, architecture = x86_64
    # gunzip mpt2sas-01.255.06.00-1-sles10sp2.x86_64.dd.gz
    # dd if=mpt2sas-01.255.06.00-1-sles10sp2.x86_64.dd of=/dev/fd0

    for SLES11, architecture = x86_64
    # gunzip mpt2sas-01.255.06.00-1-sles11.x86_64.dd.gz
    # dd if=mpt2sas-01.255.06.00-1-sles11.x86_64.dd of=/dev/fd0
```

The driver disk image can be transfered to floppy disk with the rawrite tool from dos, or the dd utility in Linux. Here is the URL for rawrite: http://www.tux.org/pub/dos/rawrite.

=== Adding Pre-Compiled Binaries to an Existing Installation ===

RPMs should be used to upgrade the driver post-install. The RPMs contain binaries for the install kernel and the released updates from Red Hat and service packs from SuSE at the time the RPM was created. The packaging provides three forms of RPMs; they are called SuSE KMP, DKMS(Dynamic Kernel Module Support), and Generic. Source RPMs are provided for the SuSE KMP and Generic RPMs. This allows one to generate binary RPM themselves for errata kernels that are released in between the normal release cycle. See the release notes file for a listing of the kernels supported by the RPM. The driver update disks should be used for architectures not supported by RPM.

Here is an example installing the generic RPMs: In this example the driver version is (01.255.06.00-1). Here is the procedure:

```
   Extract the packaging from Linux system:
   # tar -zxvf mpt2sas-release.tar.gz
   # cd rpms-1

   for RHEL5 (any Update), architecture = x86_64
   # rpm -ivh mpt2sas-01.255.06.00-1-rhel5.x86_64.rpm

   for SLES10 (any Update), architecture = x86_64
   # rpm -ivh mpt2sas-01.255.06.00-1-sles10.x86_64.rpm

   for SLES11 (any Update), architecture = x86_64
   # rpm -ivh mpt2sas-01.255.06.00-1-sles11.x86_64.rpm

   you will need to reboot for the driver to be loaded with newer version
   # reboot
```

Here is an example installing the DKMS RPMs: In this example the driver version is (01.255.06.00-1). Here is the procedure:

```
   Extract the packaging from Linux system:
   # tar -zxvf mpt2sas-release.tar.gz
   # cd dkms-1
   # tar -zxvf mpt2sas-01.255.06.00-1.dkms.tar.gz

   # install DKMS framework
   # rpm -ivh dkms-2.0.21.1-1.noarch.rpm

   # install DKMS rpm
   # rpm -ivh mpt2sas-01.255.06.00-1dkms.noarch.rpm

   you will need to reboot for the driver to be loaded with newer version
   # reboot
```

NOTE: The DKMS packaging is providing only for RHEL5(x86 and x86_64), and SLES10(x86_64), and SLES11(x86_64) pre-compiled binaries.

Here is an example of un-installing the RPM:

```
   # rpm -qa | grep mpt2sas

   look for the string having mpt2sas, and copy
   # rpm -e mpt2sas-01.255.06.00-1-rhel5
   # reboot
```

Here is an example building the Generic Binary RPM from the source RPM. In this example the driver version is (01.255.06.00-1). Here is the procedure:

```
   Extract the packaging from Linux system:
   # tar -zxvf mpt2sas-release.tar.gz
   # cd srpms-1

   # rpm -ivh mpt2sas-01.255.06.00-1.src.rpm

   for Red Hat
   # cd /usr/src/redhat/SPECS

   for SuSE
   # cd /usr/src/packages/SPECS

   build the binary
   # rpmbuild -bb mpt2sas.spec

   binary rpm located in this folder:
   # cd ../RPMS/`uname -m`
```

------ Updating Bundled Fusion-MPT Driver Source

LSI recommends that you save the original source:

```
  # tar zcvf mpt2sas.orig.tar.gz /usr/src/linux/drivers/scsi/mpt2sas
```

Continue with the instructions in "Adding Fusion-MPT Driver Source."

=== Adding or Upgrading the Fusion-MPT Source to the Linux Kernel ===

In the generic RPMs, the driver source will be placed in your installations RPM SOURCES directory; for SuSE (/usr/src/packages/SOURCES), and for Red Hat (/usr/src/redhat/SOURCES). Its also in the top folder of the packaging. In this example the driver version is (01.255.06.00-1). Here is the procedure:

```
   Extract the packaging from Linux system:
   # tar -zxvf mpt2sas-release.tar.gz
   # tar -zxvf mpt2sas-01.255.06.00.tar.gz

   copy driver source code to kernel tree
   # mkdir -p /usr/src/linux/drivers/scsi/mpt2sas
   # cp -fRv drivers/scsi/mpt2sas/* /usr/src/linux/drivers/scsi/mpt2sas
```

------ Driver Build Instructions

The following examples show how to configure and build the LSI Fusion-MPT driver(s) as kernel modules. In this example the driver version is (01.255.06.00-1). Here is the procedure to build the drivers out of kernel tree:

Extract the packaging from Linux system:

```
# tar -zxvf mpt2sas-release.tar.gz
# tar -zxvf mpt2sas-01.255.06.00.tar.gz
# cd mpt2sas
# ./compile
# ./load
```

Alternatively, here is the procedure to build driver in kernel tree

1. From the /usr/src/linux directory, ensure a clean kernel source tree by executing the following command:

```
  # make mrproper
```

2. From the /usr/src/linux directory, run the normal kernel configuration routine:

```
  # make oldconfig
or:
  # make config
or:
  # make menuconfig
or:
  # make xconfig
```

3. Here are the directions for finding the entry in menuconfig ncurses display

```
Device Drivers  --->
  SCSI device support  --->
    SCSI low-level drivers  --->
       <M>   LSI MPT Fusion SAS 2.0 Device Driver
       (128)   LSI MPT Fusion Max number of SG Entries (16 - 128) (NEW)
       [*]     LSI MPT Fusion logging facility
```

On the sub menu, select the "LSI MPT Fusion SAS 2.0 Device Driver" line, and then enter "m" to configure for building this support as a module. (Alternatively, you can enter "y" to have this support built into the kernel.)

NOTES:

- o CONFIG_SCSI_MPT2SAS_MAX_SGE: This option allows you to specify the maximum number of scatter-gather entries per I/O. The driver default is 128, which matches MAX_HW_SEGMENTS. However, it may decreased down to 16. Decreasing this parameter will reduce memory requirements on a per controller instance.
- o CONFIG_SCSI_MPT2SAS_LOGGING: This turns on a logging facility.

4. Save the kernel configuration changes. Follow any post configuration instructions, and do everything needed on your platform to rebuild the kernel. This typically includes:

```
      # make dep
      and:
        # make bzImage                        # varies on non-Intel platforms
```

5. Rebuild the kernel modules:

```
        # make modules
```

6. Optionally, (and potentially dangerous!), do everything needed on your platform to install a newly built kernel. (possibly temporarily, for sanity testing)

Be careful with this step, and be sure you know what you're doing!
It is easy to wipe out a good/stable kernel from this point forward
in the procedure!

7. (Re)Install newly compiled kernel modules:

```
        # make modules_install
```

The output from the last step should look something like this:

```
        Installing modules under /lib/modules/2.6.30/block
        Installing modules under /lib/modules/2.6.30/net
        Installing modules under /lib/modules/2.6.30/ipv4
        Installing modules under /lib/modules/2.6.30/scsi
        Installing modules under /lib/modules/2.6.30/fs
        Installing modules under /lib/modules/2.6.30/fs
        Installing modules under /lib/modules/2.6.30/cdrom
        Installing modules under /lib/modules/2.6.30/video
        Installing modules under /lib/modules/2.6.30/net
        Installing modules under /lib/modules/2.6.30/misc
```

8. Update your /boot sector with the new System.map and bzImage, re-create your ramdisk image (refer to your vendor literature), and update your boot manager--i.e., lilo.conf, grub.conf. If you are using lilo, you must run lilo -v prior to reboot.

9. Shut down the system:

Example:

```
      # shutdown -r now
```

and then reboot with the newly built Linux kernel.

=== Loading the Drivers As Modules ===

Follow the following step to load the driver binary:

Example: load the Fusion-MPT mpt2sas driver.

```
# insmod mpt2sas.ko
```

```
mpt2sas version 02.255.01.06 loaded
scsi4 : Fusion MPT SAS Host
ACPI: PCI Interrupt 0000:0b:00.0[A] -> GSI 16 (level, low) -> IRQ 177
PCI: Setting latency timer of device 0000:0b:00.0 to 64
mpt2sas0: 64 BIT PCI BUS DMA ADDRESSING SUPPORTED, total mem (6092056 kB)
mpt2sas0: PCI-MSI-X enabled: IRQ 122
mpt2sas0: iomem(0xfc47c000), mapped(0xffffc20000058000), size(16384)
mpt2sas0: ioport(0xdc00), size(256)
mpt2sas0: Allocated physical memory: size(1028 kB)
mpt2sas0: Current Controller Queue Depth(435), Max Controller Queue Depth(942)
mpt2sas0: Scatter Gather Elements per IO(128)
mpt2sas0: LSISAS2108: FWVersion(02.250.00.00), ChipRevision(0x03),
BiosVersion(00.00.00.00)
mpt2sas0: Protocol=(Initiator,Target), Capabilities=(TLR,EEDP,Snapshot Buffer, Diag Trace
Buffer, Task Set Full,NCQ)
mpt2sas0: sending port enable !!
mpt2sas0: port enable: SUCCESS
mpt2sas0: host_add: handle(0x0001), sas_addr(0x500605b0006b9310), phys(8)
```

=== Boot Setup Commands ===

1. Syntax

Setup commands can be passed to the SCSI host driver mpt2sas as a string variable using 'insmod'. The command line options can be found by typing the modinfo command.

Example:

```
# modinfo mpt2sas.ko

filename:       mpt2sas.ko
version:        02.255.01.06
license:        GPL
description:    LSI MPT Fusion SAS 2.0 Device Driver
author:         LSI Corporation <DL-MPTFusionLinux@lsi.com>
srcversion:     9D219379A3A703101318619
alias:          pci:v00001000d00000065sv*sd*bc*sc*i*
alias:          pci:v00001000d00000064sv*sd*bc*sc*i*
alias:          pci:v00001000d00000077sv*sd*bc*sc*i*
alias:          pci:v00001000d00000076sv*sd*bc*sc*i*
alias:          pci:v00001000d00000074sv*sd*bc*sc*i*
alias:          pci:v00001000d00000072sv*sd*bc*sc*i*
alias:          pci:v00001000d00000070sv*sd*bc*sc*i*
depends:        scsi_mod,scsi_transport_sas
vermagic:       2.6.16.60-0.21-debug SMP gcc-4.1
parm:           logging_level: bits for enabling additional logging info (default=0)
parm:           command_retry_count: Device discovery TUR command retry count:
(default=144) (int)
parm:           max_lun: max lun, default=16895  (int)
```

```
parm:           max_queue_depth: max controller queue depth  (int)
parm:           max_sgl_entries: max sg entries  (int)
parm:           msix_disable: disable msix routed interrupts (default=0) (int)
```

2. Available Arguments

2.1 The following command enables additional info sent to the Linux system log which can be used for troubleshooting problems. The default is to pass the logging level in hex format. Each bit is bitwise setting. Please refer to in mpt2sas_debug.h where the logging levels are defined.

Example: this enables firmware events and reply with additional info

```
#insmod mpt2sas.ko logging_level=0x218
```

Example: this enables handshake and initialization logging

```
#insmod mpt2sas.ko logging_level=0x420
```

Example: this enables application using IOCTLS logging

```
#insmod mpt2sas.ko logging_level=0x8000
```

Example: this enables manufacture configuration logging

```
#insmod mpt2sas.ko logging_level=0x800
```

Example: this enables host reset and task management logging

```
#insmod mpt2sas.ko logging_level=0x2100
```

Example: this enables task set full logging

```
#insmod mpt2sas.ko logging_level=0x80000
```

NOTE: Many of the driver debug prints are using KERN_DEBUG and KERN_INFO logging level. Red Hat and SuSE tend to set the default logging level set to a higher level, perhaps KERN_WARNING. When set to KERN_WARNING you will be missing most the debug info. To turn on the additional logs, you will need to see the set klogd to KERN_DEBUG. In both SuSE and Red Hat offer configuration of klogd from the file /etc/sysconfig/syslog. Please refer to the klogd manual page for more info.

2.2 The following command allows configuration of the command_retry_count. This tunable is for configuring the retry count for discovering devices. This is to handle some devices which report BUSY status for long duration of time.

Example: this sets the retry count to 300

```
#insmod mpt2sas.ko command_retry_count=300
```

2.3 The following command allows configuration max number of luns. The default is 511 luns. Please note that the scsi-mid layer global parameter is max_report_luns default is 511. You will need to modify max_report_luns parameter if you plan to use more than 511 luns in mpt2sas.

Example: this sets the max lun to 100

```
#insmod mpt2sas.ko max_lun=100
```

2.4 The following command allows configuration the controller queue depth. The default is 600. The maximum upper limit is set by controller firmware in facts->RequestCredit.

Example: this sets the max queue depth to 3000

```
#insmod mpt2sas.ko max_queue_depth=3000
```

2.5 The following command allows configuration the controller maximum scatter gather entries. This is maximum number of scatter-gather entries per I/O. The driver default is 128, which matches MAX_HW_SEGMENTS. However, it may decreased down to 16. Decreasing this parameter will reduce memory requirements on a per controller instance.

Example: this sets the scatter gather limit to 32

```
#insmod mpt2sas.ko max_sgl_entries=32
```

****************************** Troubleshooting **

1. Sense translation is built into the Linux kernel; providing SCSI-3 opcode string lookup and a LARGE sorted table of 463 unique SCSI-3 Additional Sense Code & Qualifier (ASC/ASCQ) strings, translated directly from a text file from the SCSI T10.org's ftp site:

ftp://ftp.t10.org/t10/drafts/spc2/asc-num.txt

Example enabling sense decoding

```
#sysctl -w dev.scsi.logging_level=0x1000
```

2.   Additional debug logging for device discovery can be enabled in the Linux kernel:

Example:

```
#sysctl -w dev.scsi.logging_level=0x1C0
```

3.   Several SCSI debug application tools are available; for example lsscsi, sdparm, SMP tools for expanders, and a variety of sg tools. These can be obtained from this URL: http://sg.danny.cz/sg. Typically these tools are provided by default in SuSE distributions.

For example, to obtain all the SAS address for your attached devices:

```
# lsscsi -t

[4:0:1:0]    disk    sas:0x5000c50000be5cf2         /dev/sdi
[4:0:2:0]    disk    sas:0x5000c50001263246         /dev/sdj
[4:0:3:0]    disk    sas:0x5000c500012632c2         /dev/sdk
[4:0:4:0]    disk    sas:0x5000c50005b04c8a         /dev/sdl
[4:0:5:0]    disk    sas:0x5000c50005b06f0e         /dev/sdm
[4:0:6:0]    disk    sas:0x5000c50005b04f3a         /dev/sdn
[4:0:7:0]    disk    sas:0x5000c50005b04d4e         /dev/sdo
[4:0:8:0]    enclosu sas:0x500605b0ffff003d         -
```

```
[4:0:9:0]    disk    sas:0x50010b900000337d         /dev/sdp
[4:0:10:0]   disk    sas:0x50010b9000002579         /dev/sdq
[4:0:11:0]   disk    sas:0x50010b900004537f         /dev/sdr
[4:0:12:0]   disk    sas:0x50010b9000029d72         /dev/sds
[4:0:13:0]   disk    sas:0x50010b900000272a         /dev/sdt
[4:0:14:0]   disk    sas:0x50010b910003389e         /dev/sdu
[4:0:15:0]   disk    sas:0x50010b91000338a6         /dev/sdv
[4:0:16:0]   disk    sas:0x50010b91000338ae         /dev/sdw
[4:0:17:0]   enclosu sas:0x500605b02222003d         -
[4:0:18:0]   disk    sas:0x5000c5000f217826         /dev/sdx
[4:0:19:0]   disk    sas:0x5000c5000f21774a         /dev/sdy
[4:0:20:0]   disk    sas:0x5000c5000f21798e         /dev/sdz
[4:0:21:0]   disk    sas:0x5000c5000f21789e         /dev/sdaa
[4:0:22:0]   disk    sas:0x5000c5000f21783e         /dev/sdab
[4:0:23:0]   disk    sas:0x5000c5000f217902         /dev/sdac
[4:0:24:0]   disk    sas:0x5000c5000f2178be         /dev/sdad
[4:0:25:0]   disk    sas:0x5000c5000f21753a         /dev/sdae
[4:0:26:0]   disk    sas:0x5000c5000f2179ca         /dev/sdaf
[4:0:27:0]   disk    sas:0x5000c5000f2177b6         /dev/sdag
[4:0:28:0]   disk    sas:0x5000c5000f2176e2         /dev/sdah
[4:0:29:0]   disk    sas:0x5000c5000f217886         /dev/sdai
[4:0:30:0]   disk    sas:0x5000c5000f2178ca         /dev/sdaj
[4:0:31:0]   disk    sas:0x5000c5000f217936         /dev/sdak
[4:0:32:0]   disk    sas:0x5000c5000f21798a         /dev/sdal
[4:0:33:0]   disk    sas:0x5000c5000f21751a         /dev/sdam
[4:0:34:0]   disk    sas:0x5000c5000f217722         /dev/sdan
[4:0:35:0]   disk    sas:0x5000c5000f2174f2         /dev/sdao
[4:0:36:0]   disk    sas:0x5000c5000f2178d2         /dev/sdap
[4:0:37:0]   disk    sas:0x5000c5000f2177a6         /dev/sdaq
[4:0:38:0]   disk    sas:0x5000c5000f21773a         /dev/sdar
[4:0:39:0]   disk    sas:0x5000c5000f2176f2         /dev/sdas
[4:0:40:0]   disk    sas:0x5000c5000f217912         /dev/sdat
[4:0:41:0]   disk    sas:0x5000c5000f2178da         /dev/sdau
[4:0:42:0]   enclosu sas:0x50080e513285003d         -
```

4. Additional scripts in the sub folder scripts are provided with the driver source code, they can be useful in obtaining detailed info pertaining to your configuration.

Extract the packaging from Linux system:

```
# tar -zxvf mpt2sas-release.tar.gz
# tar -zxvf mpt2sas-01.255.06.00.tar.gz
# cd mpt2sas/scripts
```

The hba_properties provide configuration info pertaining the host controller; the controller firmware, bios, and driver versions.

Example:

```
# ./hba_properties
```

```
host4: ioc0: fw=02.250.00.00 bios=00.00.00.00 driver=02.255.01.06 mpi=200.0b
LSISAS2108: board_name=Eval Board assembly= tracer=
nvdata_persistent=00h nvdata_default=00h
io_delay=08 device_delay=144
logging_level=00000000h
fw_queue_depth=942
sas_address=0x500605b0006b9310
```

Additional controller configuration info:

```
# ./shost_attributes

host4
board_assembly:
board_name:Eval Board
board_tracer:
cmd_per_lun:7
device_delay:144
fw_queue_depth:942
host_busy:0
host_sas_address:0x500605b0006b9310
io_delay:08
logging_level:00000000h
proc_name:mpt2sas
scan:cat: scan: Permission denied
sg_tablesize:128
state:running
uevent:cat: uevent: Permission denied
unchecked_isa_dma:0
unique_id:0
version_bios:00.00.00.00
version_fw:02.250.00.00
version_mpi:200.0b
version_nvdata_default:00h
version_nvdata_persistent:00h
version_product:LSISAS2108
```

Expander configuration info

```
# ./expander_attribute

expander-4:0
component_id:547
component_revision_id:2
component_vendor_id:LSI
level:1
product_id:Bobcat
product_rev:B0
uevent:cat: uevent: Permission denied
vendor_id:LSI CORP
```

```
 expander-4:1
 component_id:547
 component_revision_id:4
 component_vendor_id:LSI
 level:1
 product_id:Bobcat
 product_rev:0200
 uevent:cat: uevent: Permission denied
 vendor_id:LSI CORP

 expander-4:2
 component_id:531
 component_revision_id:0
 component_vendor_id:LSI
 level:1
 product_id:DE5300-SAS
 product_rev:0216
 uevent:cat: uevent: Permission denied
 vendor_id:LSI

 Device configuration info
# ./sdev_attributes

 4:0:20:0
 delete:        device_blocked:0
 iocounterbits:32
 iodone_cnt:0x26
 ioerr_cnt:0x0
 iorequest_cnt:0x26
 model:ST973402SS
 queue_depth:254
 queue_type:simple
 rescan:        retries:5
 rev:MS00
 sas_address:0x5000c5000f21798e
 sas_device_handle:0x0020
 scsi_level:6
 state:running
 timeout:60
 type:0
 uevent:        vendor:SEAGATE

 4:0:22:0
 delete:        device_blocked:0
 iocounterbits:32
 iodone_cnt:0x26
 ioerr_cnt:0x0
 iorequest_cnt:0x26
 model:ST973402SS
```

```
   queue_depth:254
   queue_type:simple
   rescan:         retries:5
   rev:MS00
   sas_address:0x5000c5000f21783e
   sas_device_handle:0x0022
   scsi_level:6
   state:running
   timeout:60
   type:0
   uevent:         vendor:SEAGATE
```

****

Fusion-MPT is a trademark of LSI Corporation
Linux is a registered trademark of Linus Torvalds
Red Hat is a registered trademark of Red Hat Software, Inc.
SuSE is a registered trademark of Novell, Inc.

****