Home Archives About

Matrix207's Blog

2013-06-20 • STORAGE

use sg_ses

##1.Introduction

The sg_ses utility enables a user “to manage and sense the state of the power supplies, cooling devices, displays, indicators, individual drives, and other non-SCSI elements installed in an enclosure”.
The SCSI Enclosure Services standards (most recent is SES-2 ANSI INCITS 448-2008) and the latest draft (ses3r03.pdf at www.t10.org) describe the format that the sg_ses utility expects to find in a SES device (“logical unit” or “process”)

##2.Command Tools

sg_map - displays mapping between linux sg and other SCSI devices

sg_ses - send controls and fetch status from a SCSI EnclosureServices (SES) device

sg = SCSI Generic
ses = SCSI Enclosure Service

##3.Use command
query mapping of expander and device name

```
[root@ ~]# sg_map
/dev/sg0  /dev/sda
/dev/sg1  /dev/sdb
/dev/sg2  /dev/sdc
/dev/sg3  /dev/sdd
/dev/sg4  /dev/sde
/dev/sg5  /dev/sdf
/dev/sg6  /dev/sdg
```

```
/dev/sg7
/dev/sg8  /dev/sdh
/dev/sg9  /dev/sdi
/dev/sg10  /dev/sdj
/dev/sg11  /dev/sdk
/dev/sg12  /dev/sdl
/dev/sg13  /dev/sdm
/dev/sg14

[root@ ~]# sg_map -i
/dev/sg0  /dev/sda  ATA       SanDisk SSD U100  10.5
/dev/sg1  /dev/sdb  ATA       Hitachi HUA72201  A3EA
/dev/sg2  /dev/sdc  ATA       Hitachi HUA72201  A39C
/dev/sg3  /dev/sdd  ATA       ST2000VX000-9YW1  CV13
/dev/sg4  /dev/sde  ATA       Hitachi HUA72201  A3EA
/dev/sg5  /dev/sdf  ATA       ST2000VX000-9YW1  CV13
/dev/sg6  /dev/sdg  ATA       ST2000VX000-9YW1  CV13
/dev/sg7  GOOXI     Bobcat            0d00
/dev/sg8  /dev/sdh  ATA       ST2000VX000-9YW1  CV13
/dev/sg9  /dev/sdi  ATA       ST2000VX000-9YW1  CV13
/dev/sg10  /dev/sdj  ATA       ST2000VX000-9YW1  CV13
/dev/sg11  /dev/sdk  ATA       ST2000VX000-9YW1  CV13
/dev/sg12  /dev/sdl  ATA       Hitachi HDE72101  A31B
/dev/sg13  /dev/sdm  ATA       Hitachi HDE72101  A3AA
/dev/sg14  GOOXI     Bobcat            0d00
```

There are two expanders, /dev/sg7 and /dev/sg14, to show the only expanders list, use below command:

```
[root@ ~]# sg_map | awk '{if($2==""){print $1}}'
/dev/sg7
/dev/sg14
```

We also can use ls to find expanders:

```
[root@ ~]# ls /dev/sg* -l
crw-rw---- 1 root disk 21,  0 Jun 18 22:54 /dev/sg0
crw-rw---- 1 root disk 21,  1 Jun 19 17:33 /dev/sg1
crw-rw---- 1 root disk 21, 15 Jun 19 23:33 /dev/sg15
crw-rw---- 1 root disk 21, 16 Jun 19 23:33 /dev/sg16
crw-rw---- 1 root disk 21, 17 Jun 19 23:33 /dev/sg17
crw-rw---- 1 root disk 21, 18 Jun 19 23:33 /dev/sg18
crw-rw---- 1 root disk 21,  2 Jun 18 22:55 /dev/sg2
crw-rw---- 1 root disk 21,  3 Jun 18 22:55 /dev/sg3
crw-rw---- 1 root disk 21,  4 Jun 19 17:33 /dev/sg4
crw-rw---- 1 root disk 21,  5 Jun 18 22:55 /dev/sg5
crw-rw---- 1 root disk 21,  6 Jun 20 17:49 /dev/sg6
```

```
crw-rw---- 1 root root 21,  7 Jun 18 22:55 /dev/sg7
crw-rw---- 1 root disk 21,  8 Jun 20 21:46 /dev/sg8
crw-rw---- 1 root disk 21,  9 Jun 20 21:46 /dev/sg9

[root@ ~]# ls /dev/sg* -l |awk '{if($4=="root"){print $10}}'
/dev/sg7
```

To view the pages of /dev/sg7

```
[root@ ~]# sg_ses -p 0 /dev/sg7
  GOOXI     Bobcat            0d00
    enclosure services device
Supported diagnostic pages:
  Supported diagnostic pages [0x0]
  Configuration (SES) [0x1]
  Enclosure status/control (SES) [0x2]
  String In/Out (SES) [0x4]
  Threshold In/Out (SES) [0x5]
  Element descriptor (SES) [0x7]
  Additional element status (SES-2) [0xa]
  Supported SES diagnostic pages (SES-2) [0xd]
  Download microcode (SES-2) [0xe]
  Subenclosure nickname (SES-2) [0xf]
```

To view the enclosure status

```
[root@ ~]# sg_ses -p 2 /dev/sg7
  GOOXI     Bobcat            0d00
    enclosure services device
Enclosure status diagnostic page:
  INVOP=0, INFO=1, NON-CRIT=0, CRIT=0, UNRECOV=0
  generation code: 0x0
  status descriptor list
    Element type: Array device slot, subenclosure id: 0
     Overall 0 descriptor:
       Predicted failure=0, Disabled=0, Swap=0, status: Unsupported
       OK=0, Reserved device=0, Hot spare=0, Cons check=0
       In crit array=0, In failed array=0, Rebuild/remap=0, R/R abort=0
       App client bypass A=0, Do not remove=0, Enc bypass A=0, Enc bypass B=0
       Ready to insert=0, RMV=0, Ident=0, Report=0
       App client bypass B=0, Fault sensed=0, Fault reqstd=0, Device off=0
       Bypassed A=0, Bypassed B=0, Dev bypassed A=0, Dev bypassed B=0
     Element 0 descriptor:
       Predicted failure=0, Disabled=0, Swap=0, status: Not installed
       OK=0, Reserved device=0, Hot spare=0, Cons check=0
       In crit array=0, In failed array=0, Rebuild/remap=0, R/R abort=0
       App client bypass A=0, Do not remove=0, Enc bypass A=0, Enc bypass B=0
```

```
        Ready to insert=0, RMV=0, Ident=0, Report=0
        App client bypass B=0, Fault sensed=0, Fault reqstd=0, Device off=0
        Bypassed A=0, Bypassed B=0, Dev bypassed A=0, Dev bypassed B=0
      Element 1 descriptor:
        Predicted failure=0, Disabled=0, Swap=0, status: OK
        OK=0, Reserved device=0, Hot spare=0, Cons check=0
        In crit array=0, In failed array=0, Rebuild/remap=0, R/R abort=0
        App client bypass A=0, Do not remove=0, Enc bypass A=0, Enc bypass B=0
        Ready to insert=0, RMV=0, Ident=0, Report=0
        App client bypass B=0, Fault sensed=0, Fault reqstd=0, Device off=0
        Bypassed A=0, Bypassed B=0, Dev bypassed A=0, Dev bypassed B=0

      ...

      Element 27 descriptor:
        Predicted failure=0, Disabled=0, Swap=0, status: Not installed
        OK=0, Reserved device=0, Hot spare=0, Cons check=0
        In crit array=0, In failed array=0, Rebuild/remap=0, R/R abort=0
        App client bypass A=0, Do not remove=0, Enc bypass A=0, Enc bypass B=0
        Ready to insert=0, RMV=0, Ident=0, Report=0
        App client bypass B=0, Fault sensed=0, Fault reqstd=0, Device off=0
        Bypassed A=0, Bypassed B=0, Dev bypassed A=0, Dev bypassed B=0
```

To view the enclosure status by specify index

```
[root@ ~]# sg_ses -p 2 -I 27 /dev/sg7
  GOOXI     Bobcat            0d00
    enclosure services device
Enclosure status diagnostic page:
  INVOP=0, INFO=1, NON-CRIT=0, CRIT=0, UNRECOV=0
  generation code: 0x0
  status descriptor list
      Element 27 descriptor:
        Predicted failure=0, Disabled=0, Swap=0, status: Not installed
        OK=0, Reserved device=0, Hot spare=0, Cons check=0
        In crit array=0, In failed array=0, Rebuild/remap=0, R/R abort=0
        App client bypass A=0, Do not remove=0, Enc bypass A=0, Enc bypass B=0
        Ready to insert=0, RMV=0, Ident=0, Report=0
        App client bypass B=0, Fault sensed=0, Fault reqstd=0, Device off=0
        Bypassed A=0, Bypassed B=0, Dev bypassed A=0, Dev bypassed B=0
```

view enclosure Additional element status
From result info, we can map the slot num with element index, also SAS address

```
[root@ ~]# sg_ses -p 0xa /dev/sg7
  GOOXI     Bobcat            0d00
    enclosure services device
```

```
Additional element status diagnostic page:
  generation code: 0x0
  additional element status descriptor list
      Element index: 0
        Transport protocol: SAS
        number of phys: 1, not all phys: 0, device slot number: 21
        phy index: 0
          device type: no device attached
          initiator port for:
          target port for:
          attached SAS address: 0x0000000000000000
          SAS address: 0x0000000000000000
          phy identifier: 0x0
      Element index: 1
        Transport protocol: SAS
        number of phys: 1, not all phys: 0, device slot number: 20
        phy index: 0
          device type: no device attached
          initiator port for:
          target port for: SATA_device
          attached SAS address: 0x500605b0000272bf
          SAS address: 0x500605b0000272a1
          phy identifier: 0x0

      ...

      Element index: 27
        Transport protocol: SAS
        number of phys: 1, not all phys: 0, device slot number: 22
        phy index: 0
          device type: no device attached
          initiator port for:
          target port for:
          attached SAS address: 0x0000000000000000
          SAS address: 0x0000000000000000
          phy identifier: 0x0

# Also, we can show the specify index additional element status
[root@ ~]# sg_ses -p 0xa -I 26 /dev/sg7
  GOOXI     Bobcat            0d00
    enclosure services device
Additional element status diagnostic page:
  generation code: 0x0
  additional element status descriptor list
      Element index: 26
        Transport protocol: SAS
        number of phys: 1, not all phys: 0, device slot number: 23
        phy index: 0
          device type: no device attached
          initiator port for:
          target port for: SATA_device
          attached SAS address: 0x500605b0000272bf
          SAS address: 0x500605b0000272ba
          phy identifier: 0x0
```

This command get a simple output:

```
# command explain:
# grep -E 'Element|slot'     search by an extended regular expression
# sed 'N;s/\n//'             join two lines together
# awk '{print $3,$15}'       printf third and 15th column

# out format: element_index slot_number
[root@ ~]# sg_ses -p 0xa /dev/sg7 |grep -E 'slot|Element' |sed 'N;s/\n//' |awk '{pri
0 21
1 20
2 16
3 12
4 24
5 25
6 26
7 27
8 8
9 4
10 0
11 1
12 3
13 2
14 7
15 6
16 5
17 11
18 10
19 9
20 15
21 14
22 13
23 19
24 18
25 17
26 23
27 22

# out format: slot_number element_index
[root@ ~]# sg_ses -p 0xa /dev/sg7 |grep -E 'slot|Element' |sed 'N;s/\n//' |awk '{pri
0 10
1 11
2 13
3 12
```

```
4 9
5 16
6 15
7 14
8 8
9 19
10 18
11 17
12 3
13 22
14 21

15 20
16 2
17 25
18 24
19 23
20 1
21 0
22 27
23 26
24 4
25 5
26 6
27 7
```

we can get the mapping relation between slot number and SAS address:

```
# 0x0000000000000000 means there is not exist physical disk
[root@ ~]# sg_ses -p 0xa /dev/sg7 |grep -E 'slot|  SAS address' |sed 'N;s/\n//' |awk
0 0x50014ee300165dde
1 0x50014ee3556bb24a
2 0x50014ee30016698a
3 0x50014ee300166666
4 0x50014ee300165dea
5 0x50014ee3aac1019e
6 0x0000000000000000
7 0x0000000000000000

8 0x0000000000000000
9 0x0000000000000000
10 0x0000000000000000
11 0x0000000000000000
12 0x0000000000000000
13 0x0000000000000000
14 0x0000000000000000
15 0x0000000000000000
16 0x500605b0000272a2
17 0x500605b0000272b9
```

```
18 0x500605b0000272b8
19 0x500605b0000272b7
20 0x500605b0000272a1
21 0x0000000000000000
22 0x0000000000000000
23 0x500605b0000272ba
24 0x500148500018fba0
25 0x500148500018fba0
26 0x0000000000000000
27 0x500148500018fba0

[root@ ~]# sg_ses -p 0xa /dev/sg7 |grep -E 'slot|  SAS address' |sed 'N;s/\n//' |awk
0 0x50014ee300165dde
1 0x50014ee3556bb24a
2 0x50014ee30016698a
3 0x50014ee300166666
4 0x50014ee300165dea
5 0x50014ee3aac1019e
16 0x500605b0000272a2
17 0x500605b0000272b9
18 0x500605b0000272b8
19 0x500605b0000272b7
20 0x500605b0000272a1
23 0x500605b0000272ba
24 0x500148500018fba0
25 0x500148500018fba0
27 0x500148500018fba0
```

##4.Control LED

Table 72 – Array Device Slot control element
(Taken from http://sg.danny.cz/sg/sg_ses.html)

```
+-Byte\Bit-+---7----+---6----+---5----+---4----+---3----+---2----+---1----+---0----+
+     0    +                          COMMON CONTROL                                +
+----------+--------+--------+--------+--------+--------+--------+--------+--------+
+          + RQST   + RQST   + RQST   + RQST   + RQST   +RQST IN +RQST    + RQST   +
+     1    + OK     + RSVD   + HOT    + CONS   + IN CRIT+FAILED  +REBUILD/+ R/R    +
+          +        + DEVICE + SPARE  + CHECK  + ARRAY  +ARRAY   +REMAP   + ABORT  +
+----------+--------+--------+--------+--------+--------+--------+--------+--------+
+     2    + RQST   + DO NOT +Reserved+ RQST   + RQST   + RQST   + RQST   +Reserved+
+          + ACTIVE + REMOVE +        +MISSING + INSERT + REMOVE + IDENT  +        +
+----------+--------+--------+--------+--------+--------+--------+--------+--------+
+     3    +     Reserved    + RQST   + DEVICE + ENABLE + ENABLE +     Reserved    +
+          +                 + FAULT  +  OFF   + BYP A  + BYP B  +                 +
+----------+--------+--------+--------+--------+--------+--------+--------+--------+
```

Insert harddisk, Blue LED light on, it is controlled by hardware.

Read/write harddisk, Blue LED flash, it is controlled by hardware also.

```
--index   : Element index (not slot index)
--set     : set   a status of specify element
--clear   : clear a status of specify element
/dev/sg5  : expander position (There is problem, how to position which disk
            belong to which expander?)

##raid failed: (Blue LED light on, green LED flash)
sg_ses --index=27 --set=1:2:1 /dev/sg5

##rebuild: (Red LED flash)
sg_ses --index=27 --set=1:1:1 /dev/sg5

##miss:
sg_ses --index=27 --set=2:4:1 /dev/sg5

##indent: (Blue and Green LED flash)
sg_ses --index=27 --set=2:1:1 /dev/sg5

##disk fault: (Red LED light on)
sg_ses --index=27 --set=fault  /dev/sg5
or
sg_ses --index=27 --set=3:5:1  /dev/sg5

all the upon command can use --clear option to clean status.

The latter three invocations use a numerical description of the field whose
format is <start_byte>:<start_bit>[:<number_of_bits>] .
The <number_of_bits> defaults to 1 when it is not given.(try to understand
it with table 72)

#clear status
sg_ses --index 27 --clear=1:1 /dev/sg5
sg_ses --index 27 --clear=1:2 /dev/sg5
sg_ses --index 27 --clear=3:5 /dev/sg5
```

Some problems:

- How can I know what color the LED would be when the command done?
- Is there exist one command that can clear all status? (I do not want to use the clear command again and again, it is too ugly. or it is not need to do these?)

##5.Other command tools

获取磁盘信息的命令工具:

```
磁盘所包含的信息

硬件信息:型号、物理类型、序列号、温度、转速、工作速率、磁盘LU、
         磁盘端口SAS地址、厂商、固件、LED灯状态等
位置信息:Enclosure、slot
逻辑信息:运行状态(在线、空闲热备盘、重构盘)、逻辑类型(成员盘、空闲热备盘、
         空闲盘)、所属RAID组、所属卷组
SMART信息



#sg_map
显示sg设备和sd设备的映射关系

#sginfo -l
获取sg设备的信息

#sg_ses -p 0xa /dev/sg18
获取磁盘槽位号、磁盘端口SAS地址

#ll  /dev/disk/by-path
获取磁盘端口SAS地址、磁盘盘符

#sg_inq  -p  0x83  /dev/sdq
磁盘LU、磁盘端口SAS地址

#udevinfo -a -p /block/sdq
获取磁盘的信息

#udevinfo -q all -n /dev/sdq
获取磁盘的详细信息

#scsi_id  -x -g -s /block/sdq
获取磁盘详细信息

#smartctl  -H /dev/sdb
查看磁盘健康状态

#smartctl -a  /dev/sdb
#smartctl -A /dev/sdb
查看磁盘SMART信息和温度

#smartctl -x  /dev/sdb
显示磁盘所有信息，包括SAS地址等
```

output status in raw format:

```
[root@ ~]# sg_ses -p 0x2 /dev/sg7 -r
    00 00 00 00 00 00 00 00  05 04 00 00 01 04 00 00
    01 01 00 00 05 00 00 00  05 00 00 00 05 00 00 00
    01 01 00 00 05 00 00 00  05 00 00 00 05 00 00 00
    05 00 00 00 05 00 00 00  05 00 00 00 05 00 00 00
    01 00 00 00 01 00 00 00  01 00 00 00 01 00 00 00
    05 00 00 00 05 00 00 00  05 00 00 00 05 00 00 00
    05 00 00 00 05 00 00 00  05 00 00 00 05 00 00 00
    05 00 00 00 11 00 00 00  11 00 00 00 11 00 00 00
    01 00 00 00 05 00 00 00

byte 1 to 8 is general description, element description if start from 8th byte.

From ses3r05.pdf Table 63 -- Element status code field (Page 70)
01    0k            Element is installed and no error conditions are known.
05    Not Installed Element is not installed in enclosure.
09~0F Reserved

So, the first 4 bytes start from 8th byte is 05 04 00 00, means not disk installed
and its status is "IN FAILED ARRAY".(See ses3r05.pdf page 77 Table 73)

The second 4 bytes is 01 04 00 00, means that disk is ok and status is "IN FAILED AR

The third 4 bytes is 01 01 00 00, means that disk ok and "R/R ABORT".

The 20th 4bytes is 11 00 00 00, means disk ok.

Notice that the frist byte 11 is equal to 01, look at Table 63, we can see
that element status code only depends on low 4 bits (0~3th bits for ELEMENT
STATUS CODE, 4th bit is SWAP, 5th is DISABLED, 6th is PRDFAIL, 7th is Reserved)
```

look ses3r05.pdf for more details:

```
Table 12 – Enclosure Status diagnostic page -- Page 30
Table 13 – Status descriptor                -- Page 32
Table 62 – Status element format            -- Page 70
Table 63 – ELEMENT STATUS CODE field        -- Page 70
Table 72 – Array Device Slot control element-- Page 76
Table 73 – Array Device Slot status element -- Page 77
```

##7.Reference

- man sg_map
- **_The sg_ses utility_**
- man sg_ses howto
- The Linux sg3_utils package
- The Linux SCSI Generic (sg) Driver
- *使用**SES**协议实现磁盘定位*
- 磁盘信息管理所需要的技术
- T10 technical Committee