



Hard disks have rotating moving parts – disks that rotate at 5400 or 7200 or more revolutions per minute – and the heads are subject to extreme acceleration and deceleration. Because they have moving parts, hard disks are subject to wear and tear. Manufacturers typically estimate a mean failure time for their products – a purely statistical value, and they provide no guarantee that the disk will not die on you within the first month. And Murphy's law says that your hard disk is more likely to die when you are not expecting it to do so, that is, you will probably lose it when you do not have a recent backup or when spare parts are hard to come by.

If you are lucky, your computer might output a message like the following when you boot it:

```
SMART Failure Predicted on⤴
Primary Master: Maxtor 34098H4
Warning! Immediately back-up ⤴
your data and replace your hard⤴
disk drive. A Failure may ⤴
be imminent.
```

The message is clear. Your hard disk looks likely to fail, and you should back up your data as soon as possible. This is a friendly warning from your computer BIOS's SMART error detection feature.

SMART Helper

SMART is the acronym for **Self-Monitoring, Analysis and Reporting Technology**,

Monitoring Hard Disks with smartmontools

Crash Prevention

Hard disks don't always die as suddenly as you might think. The right tools can help you detect hard disk issues before they become critical.

BY GABRIELE POHL

a system that all modern ATA and SCSI hard disks, as well as SCSI tape drives, should have. Besides logging measured values and errors, SMART has device testing features. Of course it is a good thing to know about an impending disaster well in advance. And this is exactly what the Smartmontools package [1] does for you. It accesses the SMART feature provided by your hard disks and runs a daemon called *smartd* to provide automated controls.

The package is available for the current versions of the Linux, FreeBSD, NetBSD, Solaris, Darwin, and even Microsoft Windows. You can download the Linux version as a binary RPM or source code version from [1], and an installation guide is available from the same address. Smartmontools supports versions 3 through 7 of the ATA/ATAPI standard. Its predecessor was a software package called Smartsuite. Development of Smartsuite was discontinued back in September 2001.

Listing 1 shows you an interactive approach to displaying basic information for a device using *smartctl*. The failed hard disk in our example is attached to

the IDE bus as the primary master; it is (or should be) accessible as */dev/hda*. These commands all require *root* privileges, since non-privileged users do not have access to device files.

Listing 1 tells you that the drive is a 34097H4 model by Maxtor; its serial number is L4101EJC; it has version YAH814Y0 of the factory software; and it complies to version 6 of the ATA/ATAPI standard. The serial number can be very important when it comes to claiming a replacement drive within the warranty period.

SMART support is enabled, as you can see in the last line of the report, which reflects the BIOS setting. If the last line tells you that SMART support is disabled, you need to enter *smartctl -s on* to enable the SMART support feature.

And How Are We Feeling Today?

Setting the *smartctl -H* flag queries the health status of a device. As you can see in Listing 2, this is FAILED for our hard disk, and that explains the warning displayed on booting the machine. The hard disk's embedded SMART logic has

Listing 1: Device Information

```
# <B>smartctl -i /dev/hda<B>
smartctl version 5.32 Copyright (C) 2002-4 Bruce Allen

=== START OF INFORMATION SECTION ===
Device Model:      Maxtor 34098H4
Serial Number:     L4101EJC
Firmware Version:  YAH814Y0
Device is:         In smartctl database [for details use: -P show]
ATA Version is:    6
ATA Standard is:   ATA/ATAPI-6 T13 1410D revision 0
Local Time is:     Tue Aug 10 12:17:11 2004 CEST
SMART support is:  Available - device has SMART capability.
SMART support is:  Enabled
```

already reallocated 637 defective sectors on the disk. This is an automatic feature that remaps damaged sectors to a small number of reserved sectors. Unfortunately, the supply of reserved sectors is now exhausted. If the hard disk had checked out okay, the result would have been PASSED.

The ATA standard defines a whole bunch of attributes that describe a device's technical properties [3], and manufacturers can add proprietary extensions. There may be some differences in the way the data are stored; attribute #9, which specifies a device's operating hours, in particular. To avoid complications smartmontools has a database with the attributes for a number of models. If your drive model is not in the latest version of the smartmontools database, you can check the smartmontools-database [2] mailing list for an update. The smartmontools homepage tells you how to do so.

The -A option tells *smartctl* to output the values of a device's attributes (see Listing 3). You can then compare the latest values with the manufacturer's threshold values. The *VALUE* column has the current value, *WORST* has the worst measured value so far, and *THRESH* the factory threshold value. A current value equal to, or below, the

```

jupiter:~ # smartctl -C /dev/hda
smartctl version 5.32 Copyright (C) 2002-4 Bruce Allen
Please page us http://smartmontools.sourceforge.net/

==== START OF READ SMART DATA SECTION ====
General SMART Values:
Offline data collection status: 0x02: Offline data collection activity
was completed without error.
Auto Offline Data Collection: Enabled.
Self-test execution status: 1 244: Self-test routine in progress...
40% of test remaining.
Total time to complete Offline
data collection: 1 430 seconds.
Offline data collection
capabilities: 0x0b: SMART execute Offline immediate.
Auto Offline data collection on/off support.
Suspend Offline collection upon new
command.
Offline surface scan supported.
Self-test supported.
No Conveyance Self-test supported.
Selective Self-test supported.
SMART capabilities: 0x0003: Saves SMART data before entering
power-saving mode.
Supports SMART auto save timer.
Error logging capability: 0x01: Error logging supported.
No General Purpose Logging support.
Short self-test routine
recommended polling time: 1 15 minutes.
Extended self-test routine
recommended polling time: 1 50 minutes.
    
```

Figure 1: SMART capability report for a disk.

threshold value indicates an error condition for the attribute in question. In our example (Listing 2), the threshold value for the *Reallocated_Sector_Ct* attribute is 63. The current value of 1 is way below that value, and this is what causes the *FAILING_NOW* message in the *WHEN_FAILED* column.

The attribute type is also important to evaluating the health state of a device. The attribute type is displayed in the

TYPE column in Listing 3, for example. *Old_age* attributes are characteristic of a normal aging process. Attributes with a *Pre-fail* tag are more serious, as they indicate an imminent failure! In this case, you should look to replace the disk in the near future.

Online and Offline Tests

SMART-enabled hard disks can collect values for specified attributes and store them without affecting performance. There are a few examples of this in Listing 3; the attributes with the *Always* label in the *UPDATED* column. The smartmontools manpage refers to this approach as an online test.

You can run *smartctl* with the -c flag to discover which SMART methods your device supports and how much time the tests take (see Figure 1). The same command indicates the test progress in the *Self-test execution status* section.

The so-called offline test can affect disk performance, although you might not notice this in real-life conditions, as the test is interrupted if the operating

Listing 2: Querying the health status

```

# <B>smartctl -H /dev/hda<B>
smartctl version 5.32 Copyright (C) 2002-4 Bruce Allen

==== START OF READ SMART DATA SECTION ====
SMART overall-health self-assessment test result: FAILED!
Drive failure expected in less than 24 hours. SAVE ALL DATA.
Failed Attributes:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE      UPDATED  WHEN_FAILED RAW_VALUE
  5 Reallocated_Sector_Ct   0x0033   001    001   063   Pre-fail Always    FAILING_NOW 637
    
```

Listing 3: Querying Attributes

```

# smartctl -A /dev/hda
smartctl version 5.32 Copyright (C) 2002-4 Bruce Allen

==== START OF READ SMART DATA SECTION ====
SMART Attributes Data Structure revision number: 16
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE      UPDATED  WHEN_FAILED RAW_VALUE
  1 Raw_Read_Error_Rate     0x000a   253    252   000   Old_age Always          -         41
  3 Spin_Up_Time            0x0027   222    222   063   Pre-fail Always          -        4458
  4 Start_Stop_Count        0x0032   253    253   000   Old_age Always          -         35
  5 Reallocated_Sector_Ct   0x0033   001    001   063   Pre-fail Always    FAILING_NOW 637
  6 Read_Channel_Margin     0x0001   253    253   100   Pre-fail Offline         -          0
  7 Seek_Error_Rate         0x000a   253    252   000   Old_age Always          -          0
  8 Seek_Time_Performance   0x0027   252    246   187   Pre-fail Always          -       38203
  9 Power_On_Minutes        0x0032   253    253   000   Old_age Always          -       16h+46m
[...]
```

system needs regular access to the disk. To launch the offline test, type `smartctl -t offline <device>`.

To start periodical offline testing – usually every four hours – you specify the `-o on` option, assuming that the manufacturer supports this feature. You can check your device's SMART capabilities to find out (see Figure 1). The *Offline data collection capabilities* should have an entry for *Auto Offline data collection on/off support*. No matter whether you run the offline test explicitly or automatically, it will update the attributes labeled as offline in the *UPDATED* column (Listing 3).

While online and offline tests only collect data, the self-diagnosis test checks the electrical and mechanical characteristics of the device and the read throughput. You can run the self-diagnosis feature while working on a machine; again, the test is interrupted if the operating system needs regular access to the device. If you prefer to disable the device for the duration of the test, you can specify the `-C` option to do so. This so-called captive mode should only be used for media that are not mounted.

A short self-diagnosis (`-t short`) takes a few minutes; a detailed self-diagnosis can take an hour or more depending on the size of the disk you are testing. To launch the extended test, launch `smartctl` with the `-t long` option.

The device stores the test results in a self-test logfile. You can tell `smartctl` to display the logfile (see Listing 4) by specifying the `-l selftest` option. Details of recent tests are displayed at the top of the list. The test automatically stops if it detects an error. The *Remaining* column then tells you how much of the test was not processed as a percentage. If the test detects a hard disk error, it stores the log-

ical address of the first defective block in the *LBA* column. As you can see in line #2 of Listing 4, this test was aborted by my typing `smartctl -X /dev/hdb`.

Besides the self-test logfile, there is also an error log that gives you details on the last five errors that have occurred. The `smartctl -l error` option displays the error log.

Automated Monitoring with *smartd*

All of the `smartctl` commands we have looked at thus far need to be launched explicitly by the admin user, and that makes them unsuitable for long-term monitoring tasks. Enter *smartd*, a daemon that automatically queries SMART attributes and performs self-tests as specified by the admin. *smartd* logs status and error messages in the syslog and optionally sends email warnings if critical errors are found.

It makes sense to launch *smartd* when you boot your machine and stop the daemon when you shutdown. To support this, smartmontools has the *smartd* shell script besides the `/usr/sbin/smartctl` and `/usr/sbin/smartd` executables. The script is typically stored below `/etc/rc.d/init.d`. You can run `chkconfig --add smartd` to set links at the required runlevels. Suse Linux users may need to move the *smartd* shell script from `/etc/rc.d/init.d` to `/etc/init.d` before doing so!

To launch *smartd* with specific options, you need to modify the `smartd_opts` variable in the script to reflect your requirements. For example, the `--interval N` option specifies the interval for launching *smartd*. The default is 1800 seconds, that is 30 minutes. You can run `smartd --help` to display your options and the manpage (`man smartd`) has more details.

smartd can be configured via entries in `/etc/smartd.conf`. If this file does not exist, or if it contains an entry for *DEVICSCAN*, *smartd* will attempt to detect the ATA and SCSI devices in your system when launched, that is, it will look for `/dev/hda` through `/dev/hdt` and `/dev/sda` through `/dev/sdz`. It then enables SMART for any devices it finds and launches into the monitoring process.

Fine Tuning with *smartd.conf*

Entries below the *DEVICSCAN* keyword specify the kind of monitoring to be performed for each device. If you do not add specific entries, the defaults apply for each device type (*ata*, *scsi*, *3ware*.) Run `man smartd.conf` for details of the defaults.

It makes sense not to monitor attributes that are subject to frequent changes, such as the temperature (attributes #194 and #231), or the operating hours (attribute #9); this ensures that your syslog stays readable. To disable these attributes, add a `-I 194 -I 231 -I 9` entry to the *smartd.conf* file. For an email alert type `-m` followed by your email address.

Mitigating the Danger of Failures

Of course you cannot prevent system failures simply by using *smartd*, as a disk can fail without any warning. But even then, hard disk monitoring can still be useful. It is not always easy to detect a failure. In a worst case scenario, the failure might go unnoticed until your next regular hard disk check, and that would mean even more loss – all the more reason to use smartmontools. ■

Listing 4: Querying the Self-Test Logfile

```
# smartctl -l selftest /dev/hdb
smartctl version 5.32 Copyright (C) 2002-4 Bruce Allen

=== START OF READ SMART DATA SECTION ===
SMART Self-test log structure revision number 1
Num Test_Description Status           Remaining LifeTime LBA_of_
                                (hours) first_error
# 1 Extended offline Completed without error 00%      453      -
# 2 Extended offline Aborted by host       90%      451      -
# 3 Short offline   Completed without error 00%      451      -
```

INFO

- [1] smartmontools homepage:
<http://smartmontools.sourceforge.net/>
- [2] smartmontools-database mailing list
homepage: <https://lists.sourceforge.net/lists/listinfo/smartmontools-database>
- [3] Guide to SMART attributes:
<http://freepgs.com/smart/attributes.php>

THE AUTHOR

Gabriele Pohl is an Oracle DBA, Linux administrator, IT coach, and consultant (<http://www.dipohl.com/>). Thanks to Steffen Grunewald and Bruce Allen for their support!