Smartmontools

Introduction

Smartmontools is a set of applications that can test hard drives and read their hardware SMART statistics. Note: SMART data may not accurately predict future drive failure, however abnormal error rates may be an indication of possible hardware failure or data inconsistency.

This how to will help you to configure Smartmontools to do actions such as shut down the computer or send an e-mail when the disk is going to fail.

Prerequisites

- A modern S.M.A.R.T. capable hard disk

Setting up

Installation

You can install the `smartmontools` package from the Synaptic Package Manager (see SynapticHowto), or by typing the following into the terminal:

```
sudo apt-get install smartmontools
```

Checking a drive for SMART Capability

To ensure that your drive supports SMART, type:

```
sudo smartctl -i /dev/sda
```

where /dev/sda is your hard drive. This will give you brief information about your drive. The last two lines may look something like this:

```
SMART support is: Available - device has SMART capability.
SMART support is: Enabled
```

Enabling SMART

In the case that SMART is not enabled for your drive, you can enable it by typing:

```
sudo smartctl -s on /dev/sda
```

Testing a Drive

You may run any type of test while the drive is mounted although there may be some drop in performance. There are three types of test that can be conducted on a drive:

1. Short
2. Extended (Long)
3. Conveyance

To find an estimate of the time it takes to conduct each test, type:

```
sudo smartctl -c /dev/sda
```

The most useful test is the extended test (long). You can initiate the test by typing:

```
sudo smartctl -t long /dev/sda
```

Results

You can view a drive's test statistics by typing:

```
sudo smartctl -l selftest /dev/sda
```

To display detailed SMART information for an IDE drive, type:

```
sudo smartctl -a /dev/sda
```

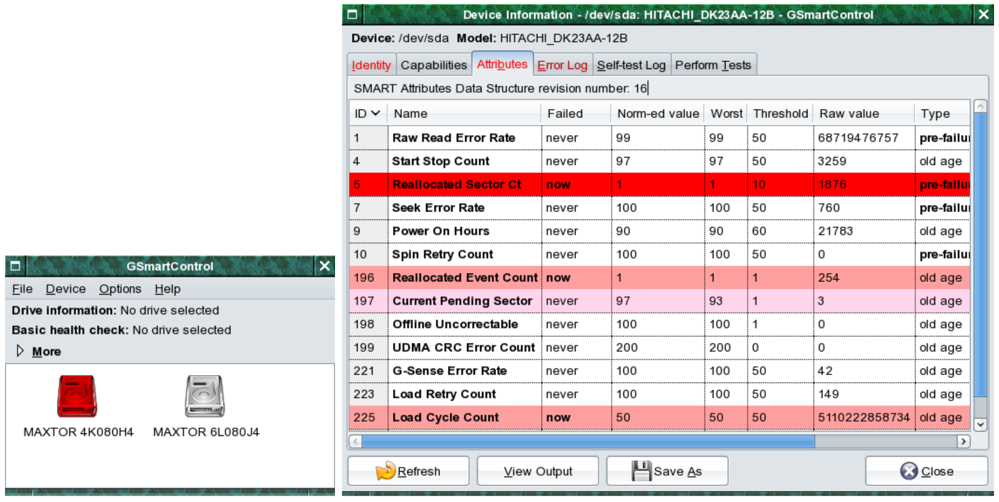
To display detailed SMART information for a SATA drive, type:

```
sudo smartctl -a -d ata /dev/sda
```

Note: This also works for IDE drives in new kernels that are being run through the SCSI stack and show up as /dev/sdX

Suggested application: GSmartControl

Take a look at GSmartControl. It's a nice graphical frontend to smartctl; it shows all SMART values, and highlights those that indicate old age or impending failure, plus you may run tests on demand:



Drive information: No drive selected
Basic health check: No drive selected

Device: /dev/sda **Model:** HITACHI_DK23AA-12B

SMART Attributes Data Structure revision number: 16

ID	Name	Failed	Norm-ed value	Worst	Threshold	Raw value	Type
1	**Raw Read Error Rate**	never	99	99	50	68719476757	**pre-failu**
4	**Start Stop Count**	never	97	97	50	3259	old age
5	**Reallocated Sector Ct**	**now**	1	1	10	1876	**pre-failu**
7	**Seek Error Rate**	never	100	100	50	760	**pre-failu**
9	**Power On Hours**	never	90	90	60	21783	old age
10	**Spin Retry Count**	never	100	100	50	0	**pre-failu**
196	**Reallocated Event Count**	**now**	1	1	1	254	old age
197	**Current Pending Sector**	never	97	93	1	3	old age
198	**Offline Uncorrectable**	never	100	100	1	0	old age
199	**UDMA CRC Error Count**	never	200	200	0	0	old age
221	**G-Sense Error Rate**	never	100	100	50	42	old age
223	**Load Retry Count**	never	100	100	50	149	old age
225	**Load Cycle Count**	**now**	50	50	50	5110222858734	old age

As usual, you may install it from Synaptic or running `sudo apt-get install gsmartcontrol`.

Advanced: Running as Smartmontools as a Daemon

You can run Smartmontools in the background and have it check drives and email when there are issues:

Open the file `/etc/default/smartmontools` with your favourite text editor. For example (using vim): `sudo vim /etc/default/smartmontools`. Uncomment the line `start_smartd=yes`.

How `smartd` is going to scan the disks and what it will do in case of errors is controlled by the daemon configuration file, `/etc/smartd.conf`. Again, use your favourite text editor to open this file. There should be one uncommented line, similar to:

```
DEVICESCAN -m root -M exec /usr/share/smartmontools/smartd-runner
```

In this example (which is the default for Karmic), `smartd` will:

- scan for all ATA/SCSI devices (`DEVICESCAN`). The rest of the file will be ignored;
- mail a report to the 'root' account in case of trouble (`-m`);
- but instead of the `mail` command, it will execute `/usr/share/smartmontools/smartd-runner` and feed the report to it (`-M exec program`).

`/usr/share/smartmontools/smartd-runner` is a script that basically saves the report to a temporary file, and then runs anything it finds in `/etc/smartmontools/run.d/`; take a look there to understand what you already have (there should be a script that mails the report).

There are several `-M` directives that change when and how often reports are sent. You need to specify (`-m something`) in order to use them, even if you're not sending any mail.

You may include some useful options:

```
DEVICESCAN -H -l error -l selftest -f -s (O/../../5/11|L/../../5/13|C/../../5/15) -m root -M exec /usr/share/smartmontools/smartd-runner
```

In this example, `smartd` will:

- check the SMART health status (`-H`);
- report increases in both SMART error logs (`-l`);
- check for failure of any Usage Attributes (`-f`);
- schedule an Offline Immediate Test every Friday at 11 am, a Long Self-Test every Friday at 1 pm, and a Conveyance Self-Test every Friday at 3 pm (`-s`) -- see the `smartd` manual page for what these tests do so you can choose what suits you.

You may also replace DEVICESCAN with the path of the device which you'd like to be monitored (e.g. `/dev/sda`), and the daemon will only monitor this drive. You'll need one such line for each device.

Actions in case of trouble

You'll want to configure the actions `smartd` will take in case of trouble. If all you want is a notification shown on your desktop, skip to "Personal computer" below.

Most of the time, you only need to place a script in `/etc/smartmontools/run.d/`. Whenever `smartd` wants to send a report, it will execute `smart-runner` and the latter will run your script.

You have several variables available to your script (again, see the `smartd` manpage). These come from a test run:

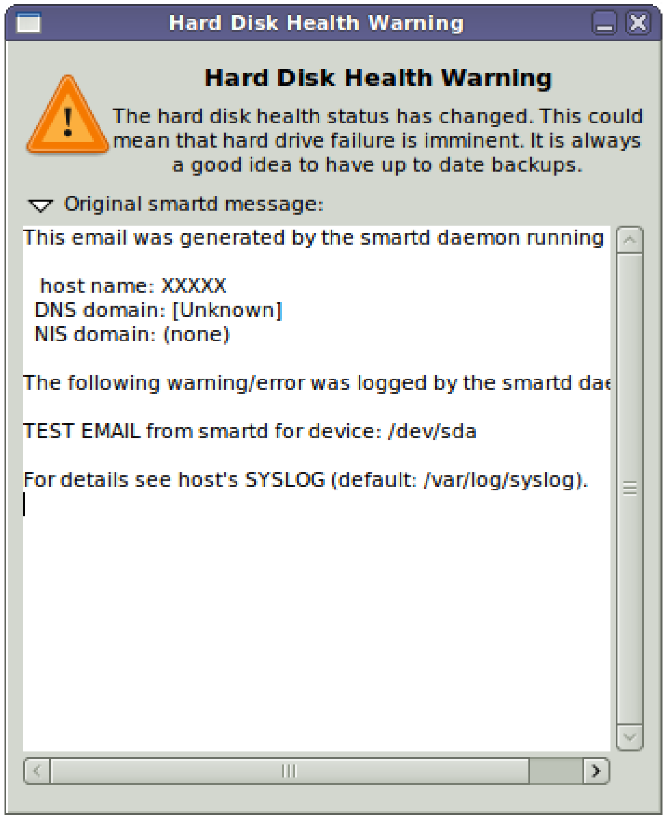
```
SMARTD_MAILER=/usr/share/smartmontools/smartd-runner
SMARTD_SUBJECT=SMART error (EmailTest) detected on host: XXXXX
SMARTD_ADDRESS=root
SMARTD_TFIRSTEPOCH=1267409738
```

```
SMARTD_FAILTYPE=EmailTest
SMARTD_TFIRST=Sun Feb 28 21:45:38 2010 VET
SMARTD_DEVICE=/dev/sda
SMARTD_DEVICETYPE=sat
SMARTD_DEVICESTRING=/dev/sda
SMARTD_FULLMESSAGE=This email was generated by the smartd daemon running on:
SMARTD_MESSAGE=TEST EMAIL from smartd for device: /dev/sda
```

Your script also has a *temporary* copy of the report available as "$1". It will be deleted after you finish but the same content is written to `/var/log/syslog`.

Personal computer

For a visual notification, you may just install `smart-notifier`. You will see a large popup with the report:



Hard Disk Health Warning

The hard disk health status has changed. This could mean that hard drive failure is imminent. It is always a good idea to have up to date backups.

```
This email was generated by the smartd daemon running

 host name: XXXXX
 DNS domain: [Unknown]
 NIS domain: (none)

The following warning/error was logged by the smartd dae

TEST EMAIL from smartd for device: /dev/sda

For details see host's SYSLOG (default: /var/log/syslog).
```

Alternatively, you may create a custom notification (bubble) as seen in other GNOME programs.

You will need to install the `libnotify-bin` package:

```
sudo aptitude install libnotify-bin
```

Now create a text file called `60notify` in `/etc/smartmontools/run.d`:

```
sudo vi /etc/smartmontools/run.d/60notify
```

and add the following to the file:

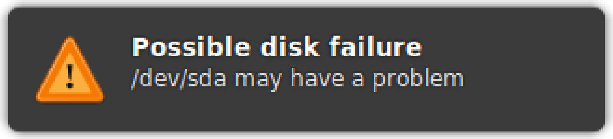
```
DISPLAY=:0.0 notify-send --icon=important "Possible disk failure" "$SMARTD_DEVICE may have a problem"
```

(The `DISPLAY=:0.0` part is a variable assignment that helps programs to locate your X server. It's already set for your terminal, but the script lacks it since it is being run inside a different session).

Now give it execute permissions:

```
sudo chmod +x /etc/smartmontools/run.d/60notify
```

This will produce a nice libnotify bubble with a warning icon:



Possible disk failure
/dev/sda may have a problem

You may also experiment with Zenity:

```
DISPLAY=:0.0 zenity --text-info --filename="$1" --title="smartd: $SMARTD_DEVICE may have a problem"
```

Notice: Be very careful with these scripts as they are run under the root account.

Server

Here, you may wish to handle things differently. In this example we want to mail an admin *and* shut down the server. Comment out the line that contains `DEVICESCAN`, by adding # to the beginning of the line. Then, add this to the end of the file:

```
/dev/hda -H -l error -l selftest -f -s (O/../../5/11|L/../../5/13|C/../../5/15) \
-m admin@somewhere.com -M exec /usr/share/smartmontools/smartd-runner
```

(Be sure not to add any whitespace after the "\")

Now, we are going to make the script which is going to shut down the computer *after* we mail the admin. Create a text file called `99shutdown` in `/etc/smartmontools/run.d` and add the following to the file:

```
sleep 40
shutdown -h now
```

The number *99* at the start of the filename is to ensure that it is called last when `smartd-runner` runs. It will wait 40 seconds and then shut down the computer. Of course, you may customize this at will; you may not wish to turn off the server.

Now, it is time to start the daemon:

```
sudo service smartmontools start
```

Testing

If you want to test all these actions, add `-M test` after `exec /usr/share/smartmontools/smartd-runner` and restart the daemon (`sudo service smartmontools restart`). When the daemon comes up, it will execute the script immediately with a test message. **Notice:** If you included the `shutdown -h` line, the script will shut down the computer as soon as the service starts. To fix this, you will have to start the computer in recovery mode and remove the `-M test` option from `/etc/smartd.conf`.

Based on Gentoo Wiki: HOWTO Monitor your hard disk(s)withsmartmontools.

Note

Before running this, be sure to check that you have a "mail" command, and do a test first to your address. On my default Fiesty:

jim@beorn:~$ mail

The program 'mail' can be found in the following packages:

- mailx
- mailutils

Try: sudo apt-get install <selected package>

Make sure you have the 'universe' component enabled

bash: mail: command not found

Utility: Checking all disks at once

Note: Following the Gentoo Wiki I made a modified script which checks all the disk in `/dev/disk/by-id/` Just invoke the script below as follows:

```
./smart.sh short|long|offline
```

The script creates a directory named `smart-logs` and stores all the files there.

```
# Script by Meliorator. irc://irc.freenode.net/Meliorator
# modified by Ranpha
[ ! "$@" ] && echo "Usage: $0 type [type] [type]"

[ ! -e smart-logs ] && mkdir smart-logs
[ ! -d smart-logs ] && Can not create smart-logs dir && exit 1

a=0

for t in "$@"; do

        case "$t" in
                offline)  l=error;;
                short|long)  l=selftest;;
                *) echo $t is an unrecognised test type. Skipping... && continue
        esac

       for hd in  /dev/disk/by-id/ata*; do
                r=$(( $(smartctl -t $t -d ata $hd | grep 'Please wait' | awk '{print $3}') ))
                echo Check $hd - $t test in $r minutes
                [ $r -gt $a ] && a=$r
       done
     echo "Waiting $a minutes for all tests to complete"
                sleep $(($a))m

        for hd in /dev/disk/by-id/ata*; do
                smartctl -l $l -d ata $hd 2>&1 >> smart-logs/smart-${t}-${hd##*/}.log
        done


done

for i in {1..10}; do
        sleep .01
        echo -n -e \\a
done

echo "All tests have completed"
```

(Remember to give execute permissions to the script with `chmod +x smart.sh`).

CategoryHardware

Smartmontools (last edited 2015-12-02 18:26:54 by Bernd @ 194-166-42-118.adsl.highway.telekom.at[194.166.42.118]:Bernd)