NEW HARDWARE TESTING METHODOLOGY AT IHEP DATA CENTER

V. Ezhova[a], V. Kotliar

Institute for High Energy Physics named by A.A. Logunov of National Research Centre

"Kurchatov Institute", Nauki Square 1, Protvino, Moscow region, Russia, 142281

E-mail: [a] victoria.ezhova@ihep.ru

The modern computing center is not only about a production capacity, it is also about a stable work. Stability means not only software, the reliability of the software component but also assembly components. It is important to ensure this before the production. One of the ways is testing components for performance, reliability, assembly defects. In this work we will present the methodology for identifying problem points, the specific tests and some results of their work based on IHEP data center.

Keywords: data center, disk system, RAID, disk server, software, assembly components.

Victoria Ezhova, Viktor Kotliar

1. Introduction

IHEP has a data center which consists of computing nodes and data storage systems. Every day IHEP central computing cluster produce thousands of calculations related to research activities, both IHEP and GRID experiments. These are data from experiments ATLAS, ALICE, CMS, LHCb. This talking about the safety of the results of calculations. So each machine must meet the reliability requirements. Each new server gets a series of checks which help to identify some problems even before launching into production. During the testing phase, it is possible to adjust the settings and to identify some vulnerabilities, including incompatibility of components or a physical defect.

In this article it will talk about some methods of checking servers before starting up, additional monitoring tools.

2. RAM check

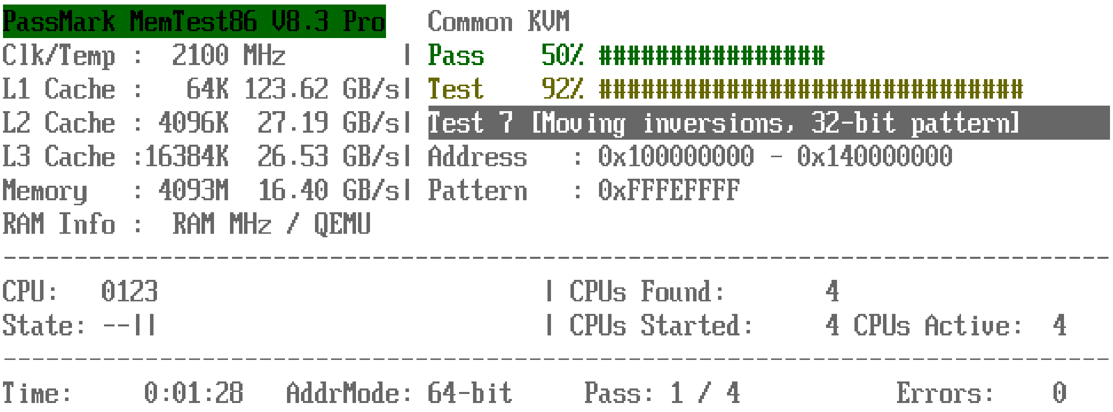
Bad RAM is one of the most frustrating computer problems as the symptoms are often random and difficult to identify.

- RAM is not compatible with motherboard
- there is a defect in some slats
- errors appear only during heating (long-term operation)

In the case of complex technical components, the problems not with the fact that the memory does not work at all. As fact - it may not work with certain conditions. The unstable operation of the RAM can begin after it "warms up", or when the load increases. In this case, the temperature rises, and the higher the temperature, the more unstable the memory works.

The main task of memory testing programs is to load the RAM as much as possible. This will raise the temperature and allow to check the stability of operation in this mode. The problem is that not all programs are able to load the RAM to the maximum. This is exactly what MemTest86 succeeds [1].

Memtest boots from a USB flash drive [fig. 1]. It executes a series of numbered test sections to check for errors. These test sections consist of a combination of test algorithm, data pattern and cache setting. The execution order for these tests were arranged so that errors will be detected as rapidly as possible [2].



```
PassMark MemTest86 V8.3 Pro   Common KVM
Clk/Temp :  2100 MHz        | Pass    50% ################
L1 Cache :   64K 123.62 GB/s| Test    92% ##############################
L2 Cache : 4096K  27.19 GB/s| Test 7 [Moving inversions, 32-bit pattern]
L3 Cache :16384K  26.53 GB/s| Address   : 0x100000000 - 0x140000000
Memory   : 4093M  16.40 GB/s| Pattern   : 0xFFFEFFFF
RAM Info :  RAM MHz / QEMU
------------------------------------------------------------------------------
CPU:   0123                        | CPUs Found:      4
State: --||                        | CPUs Started:    4 CPUs Active:  4
------------------------------------------------------------------------------
Time:     0:01:28   AddrMode: 64-bit     Pass: 1 / 4          Errors:    0
```

Figure 1. Memtest work screen

The program writes information to each memory block, and then reads it and checks for errors. During testing, the utility makes several passes, which allows it to identify and compile a list of bad memory blocks in the BadRAM format. It runs in a circle. The special attention should be paid to the Errors and Walltime blocks. And as soon as one circle is completed, the message "***** Pass complete, (no, 1, 2, 10 ...) errors, press Esc to exit *****" will appear at the bottom of the screen.

3. CPU check

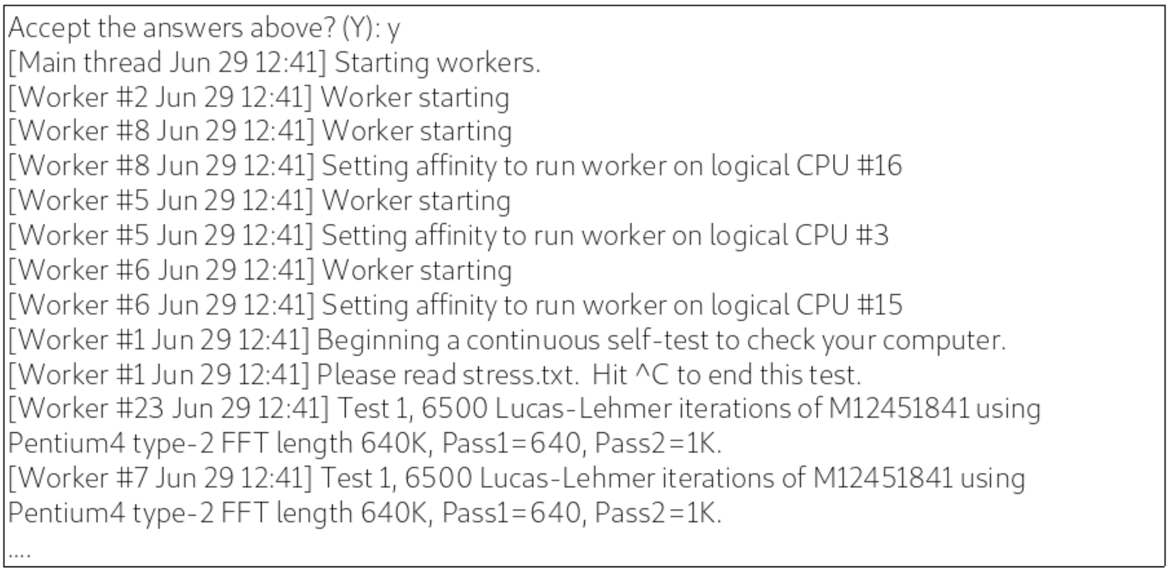
Here we will focus on testing the processor for speed and stability in extreme mode. Mprime fits well for these purposes. It is a freeware application. It is used to check stability when overclocking computers. Overclocking is the practice of increasing the speed of the CPU and/or memory to make a machine faster at little cost. Typically, overclocking involves pushing a machine past its limits and then backing off just a little bit. The principle of operation is based on a large amount of mathematical calculations.

This program is a good stress test for the CPU, memory, L1 and L2 caches, CPU cooling, and case cooling. The torture test runs continuously, comparing your computer's results to results that are known to be correct. Any mismatch and you've got a problem [3].

There are several options for the torture test (menu option 15).

- Small FFTs (option 1) to stress the CPU
- In-place large FFTs (option 2) to test the CPU and memory controller
- Blend (option 3) is the default and constitutes a hybrid mode which stresses the CPU and RAM.

Below is a part of the script execution [fig. 2].



```
Accept the answers above? (Y): y
[Main thread Jun 29 12:41] Starting workers.
[Worker #2 Jun 29 12:41] Worker starting
[Worker #8 Jun 29 12:41] Worker starting
[Worker #8 Jun 29 12:41] Setting affinity to run worker on logical CPU #16
[Worker #5 Jun 29 12:41] Worker starting
[Worker #5 Jun 29 12:41] Setting affinity to run worker on logical CPU #3
[Worker #6 Jun 29 12:41] Worker starting
[Worker #6 Jun 29 12:41] Setting affinity to run worker on logical CPU #15
[Worker #1 Jun 29 12:41] Beginning a continuous self-test to check your computer.
[Worker #1 Jun 29 12:41] Please read stress.txt.  Hit ^C to end this test.
[Worker #23 Jun 29 12:41] Test 1, 6500 Lucas-Lehmer iterations of M12451841 using
Pentium4 type-2 FFT length 640K, Pass1=640, Pass2=1K.
[Worker #7 Jun 29 12:41] Test 1, 6500 Lucas-Lehmer iterations of M12451841 using
Pentium4 type-2 FFT length 640K, Pass1=640, Pass2=1K.
....
```

Figure 2. The script execution

The archive contains a script and some information files. At the end of the test, 2 more files are formed:

- prime.txt - a configuration file containing information about the GUI's output behaviour and the connection with the PrimeNet server;
- results.txt - a history file containing completed workunits and related stuff.

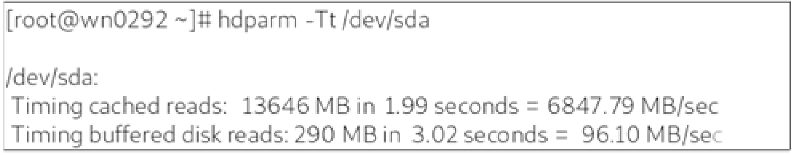
Errors will be reported if only they occur both to stdout and to ~/results.txt for review later. Many people do not consider a system as 'stable' unless it can run the Large FFTs for a 24 hour period [4].

This stress testing software is classified as high power for electrical power. It is recognized universally as one defacto measure of system stability.

4. Test HDD speed

How to find out what maximum speed of hard drive supports? One option is to use the hdparm or dd command. It provides a command line interface to various hard disk ioctls supported by the stock Linux ATA/IDE/SATA device driver subsystem.

Hdparm can set parameters such as drive caches, sleep mode, power management, acoustic management, and DMA settings. Here is example to measure hard disk data transfer speed [fig. 3].

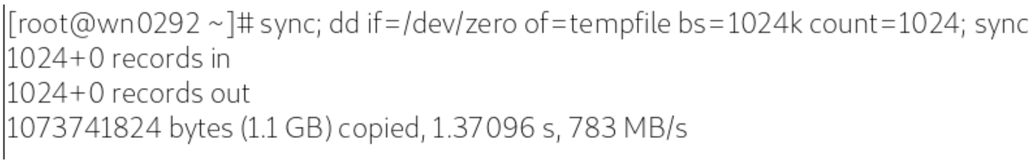


```
[root@wn0292 ~]# hdparm -Tt /dev/sda

/dev/sda:
 Timing cached reads:  13646 MB in 1.99 seconds = 6847.79 MB/sec
 Timing buffered disk reads: 290 MB in  3.02 seconds =  96.10 MB/sec
```

Figure 3. Hard disk data transfer speed

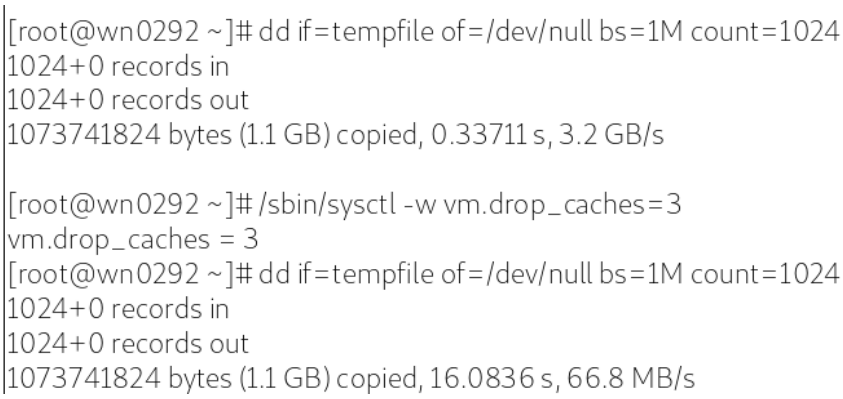
The second utility also can check write and read speed [fig. 4].



```
[root@wn0292 ~]# sync; dd if=/dev/zero of=tempfile bs=1024k count=1024; sync
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB) copied, 1.37096 s, 783 MB/s
```

Figure 4. Write and read speed

Tempfile has been cached into a buffer and its read speed will be faster than real. To get real speed, it need to clear the cache [fig. 5] [5].



```
[root@wn0292 ~]# dd if=tempfile of=/dev/null bs=1M count=1024
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB) copied, 0.33711 s, 3.2 GB/s

[root@wn0292 ~]# /sbin/sysctl -w vm.drop_caches=3
vm.drop_caches = 3
[root@wn0292 ~]# dd if=tempfile of=/dev/null bs=1M count=1024
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB) copied, 16.0836 s, 66.8 MB/s
```

Figure 5. Speed without the cache

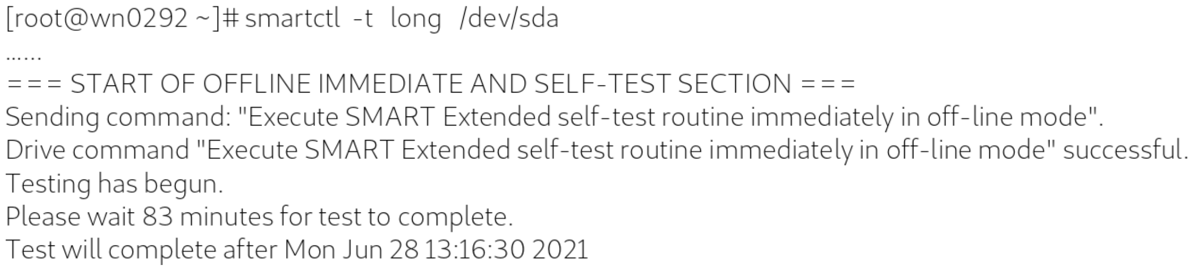
5. Smartmontools HDD check

To get all the available SMART information about a storage device, we can launch the smartmontools utility with the -a option (short for -all) and of course pass the path of the device we want to check as the last argument of the command.

The smartctl utility can be used to launch a variety of self-tests:

- short
- long [fig. 6]
- conveyance (ATA devices only)
- select (ATA devices only)

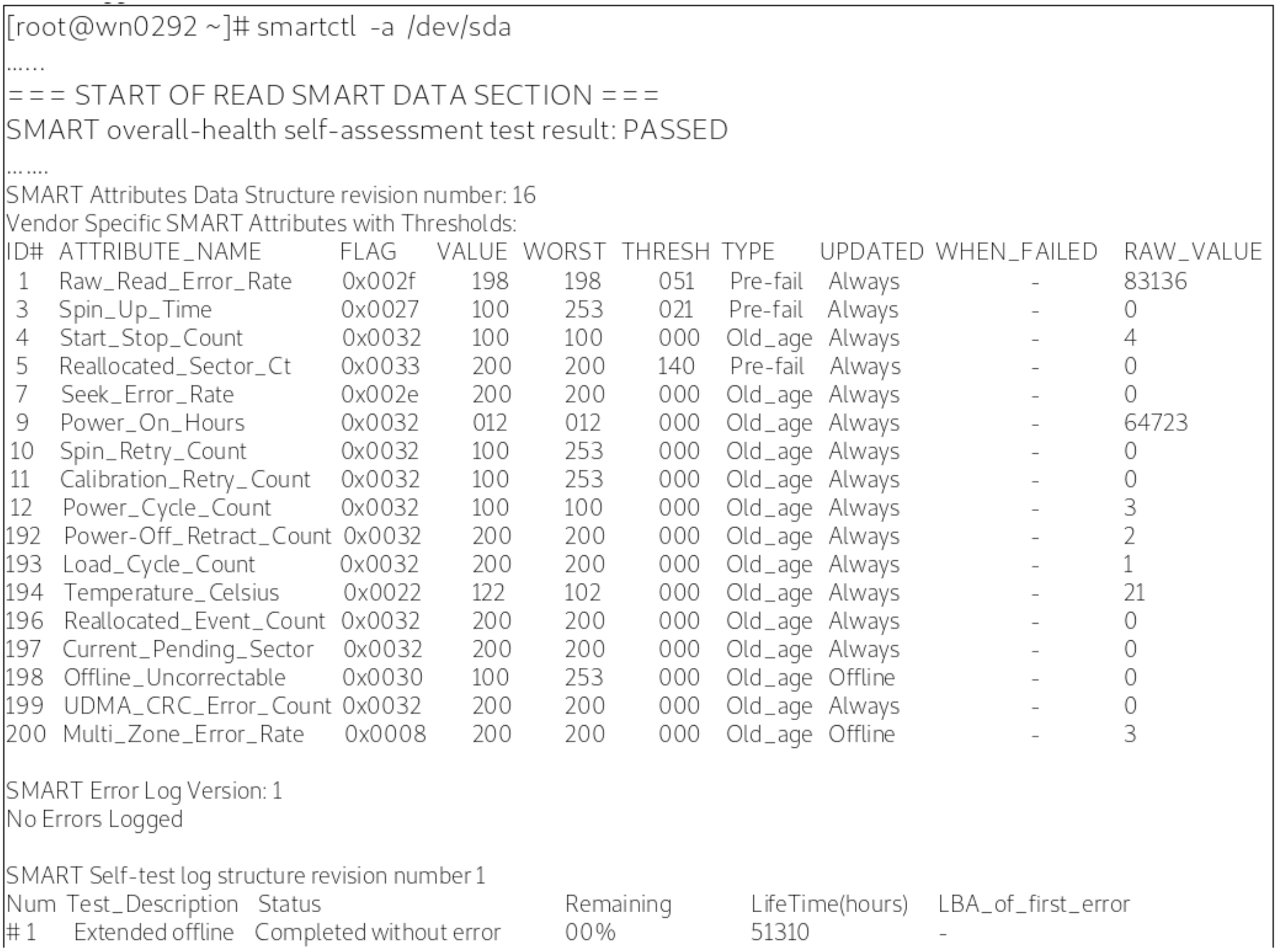
In our case the second type is used. The long test is basically a more accurate version of the “short” test. It can take a lot of time to complete: as stated in the the smartctl manual, it can last from tens of minutes to several hours [6].



```
[root@wn0292 ~]# smartctl  -t  long  /dev/sda
……
=== START OF OFFLINE IMMEDIATE AND SELF-TEST SECTION ===
Sending command: "Execute SMART Extended self-test routine immediately in off-line mode".
Drive command "Execute SMART Extended self-test routine immediately in off-line mode" successful.
Testing has begun.
Please wait 83 minutes for test to complete.
Test will complete after Mon Jun 28 13:16:30 2021
```

Figure 6. The long test execution

To check the current status of the /dev/sda device [fig. 7]



```
[root@wn0292 ~]# smartctl  -a  /dev/sda
……
=== START OF READ SMART DATA SECTION ===
SMART overall-health self-assessment test result: PASSED
…….
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     0x002f   198   198   051    Pre-fail  Always       -       83136
  3 Spin_Up_Time            0x0027   100   253   021    Pre-fail  Always       -       0
  4 Start_Stop_Count        0x0032   100   100   000    Old_age   Always       -       4
  5 Reallocated_Sector_Ct   0x0033   200   200   140    Pre-fail  Always       -       0
  7 Seek_Error_Rate         0x002e   200   200   000    Old_age   Always       -       0
  9 Power_On_Hours          0x0032   012   012   000    Old_age   Always       -       64723
 10 Spin_Retry_Count        0x0032   100   253   000    Old_age   Always       -       0
 11 Calibration_Retry_Count 0x0032   100   253   000    Old_age   Always       -       0
 12 Power_Cycle_Count       0x0032   100   100   000    Old_age   Always       -       3
192 Power-Off_Retract_Count 0x0032   200   200   000    Old_age   Always       -       2
193 Load_Cycle_Count        0x0032   200   200   000    Old_age   Always       -       1
194 Temperature_Celsius     0x0022   122   102   000    Old_age   Always       -       21
196 Reallocated_Event_Count 0x0032   200   200   000    Old_age   Always       -       0
197 Current_Pending_Sector  0x0032   200   200   000    Old_age   Always       -       0
198 Offline_Uncorrectable   0x0030   100   253   000    Old_age   Offline      -       0
199 UDMA_CRC_Error_Count    0x0032   200   200   000    Old_age   Always       -       0
200 Multi_Zone_Error_Rate   0x0008   200   200   000    Old_age   Offline      -       3

SMART Error Log Version: 1
No Errors Logged

SMART Self-test log structure revision number 1
Num  Test_Description    Status                  Remaining  LifeTime(hours)  LBA_of_first_error
# 1  Extended offline    Completed without error       00%      51310         -
```

Figure 7.Result of long test

The command above produces a lot of output. Among the other things, we can see the status of various SMART parameters. Very important parameters to check are, among the others, “Reallocated_Sector_Ct” and “Current_Pending_Sector”. In both cases if the RAW_VALUE is something other than 0, we should be very careful and start to backup data on the hard drive. The Reallocated_Sector_Ct is the count of sectors on the block device which cannot be used correctly.

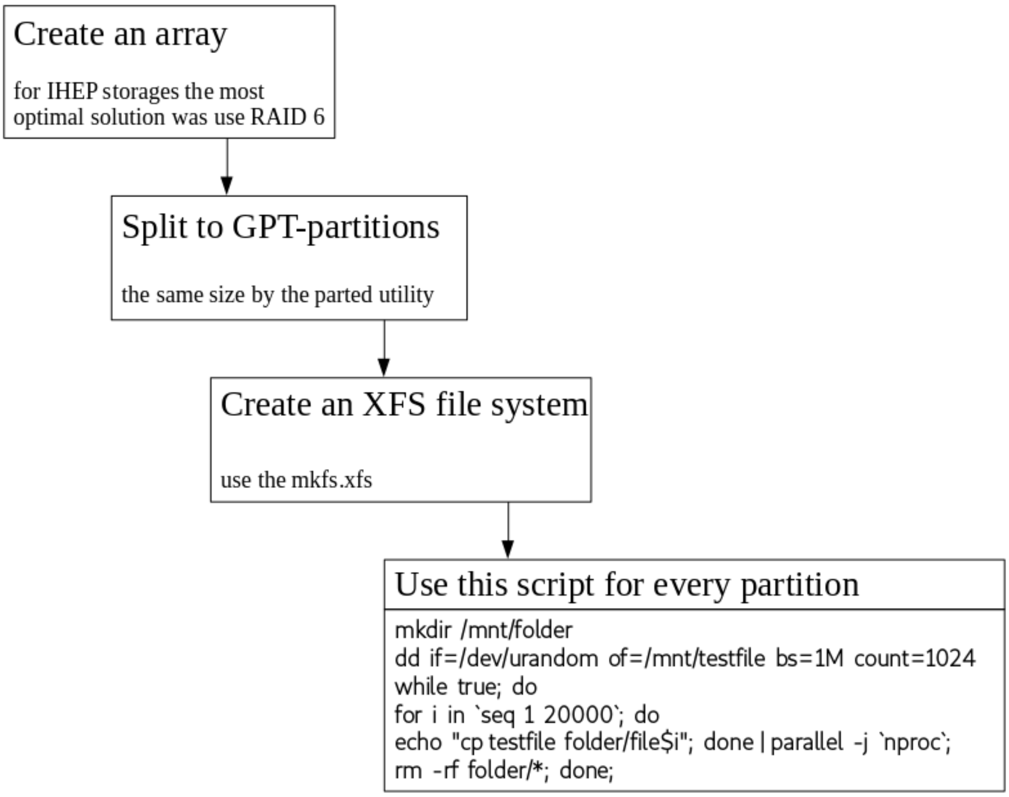
When such a sector is found it is remapped to one of the available spare sectors of the storage device, and data contained in it is relocated. The Current_Pending_Sector attribute, instead, is the count of bad sectors that are still waiting to be remapped [6].

6. RAID Controller

Storage servers work with a huge volume of data. So the safety of information is very important. For IHEP storages the most optimal solution was to use RAID 6. Failure is simulated after assembly of the array during active server operation. Next the disk will be replace and after the file system will be checked.

Here we will talk about the hardware controller. IHEP datacenter uses Adaptec and 3ware RAID controllers. The use of a hardware RAID controller is justified in most cases when high speed and reliability of the disk subsystem is required.

At first the array will be splitted into GPT-partitions of the same size by the parted utility. The best solution will be to use mkfs.xfs to create the XFS file system. The XFS filesystem is a high performance journalling filesystem. A good idea is to enable write cache, this will speed up the rebuild process (at least for RAID 6).The procedure is shown below [fig. 8].


Create an array
for IHEP storages the most optimal solution was use RAID 6
Split to GPT-partitions
the same size by the parted utility
Create an XFS file system
use the mkfs.xfs
Use this script for every partition

```
mkdir /mnt/folder
dd if=/dev/urandom of=/mnt/testfile bs=1M count=1024
while true; do
for i in `seq 1 20000`; do
echo "cp testfile folder/file$i"; done | parallel -j `nproc`;
rm -rf folder/*; done;
```

Figure 8. File system integrity check

The number 20,000 is taken from the disk array - 20 TB [7]. Also use the xfs_repair command to repair an XFS file system. Run the command xfs_repair -n to see which bugs can be fixed (but not fix them).

If no errors were found, the test was successful.

7. Conclusions

The article presents the methodology for identifying problem points, software tests and specific tests that are used by IHEP. Results of their work are based on IHEP data center. At this stage, components are tested for performance, reliability, assembly defects. It does not include connection. This year a gigabit channel was connected to IHEP. Signal availability is enough now, but in the future it is planned to introduce network bandwidth tests with the advent of a 10 GB channel.

References

[1] The article about memtest. Available at: http://reanimator.mixed-spb.ru/memtest/what-is-memtest.php (accessed 30.06.2021)

[2] «Memtest86 technical information». Available at: https://www.memtest86.com/tech_individual-test-descr.html (accessed 30.06.2021)

[3] «Free Mersenne Prime Search Software» (Prime95 Version 30.3 build 6). Available at: https://www.mersenne.org/download/ (accessed 30.06.2021)

[4] «Stress testing software». Available at: https://wiki.archlinux.org/title/Stress_testing (accessed 30.06.2021)

[5] «Test HDD speed in Linux» Available at: https://wdocs.readthedocs.io/hdd_speed_test.html (accessed 30.06.2021)

[6] «How to check an hard drive health from the command line using smartctl». Available at: https://linuxconfig.org/how-to-check-an-hard-drive-health-from-the-command-line-using-smartctl (accessed 30.06.2021)

[7] «The disk server testing» (dp0031 — lustre). Available at: https://redmine.ihep.su/projects/lpoit/knowledgebase/articles/5 (accessed 30.06.2021)