

## Re: bad blocks on raid5 cause filesystem failure

**Source:** <http://linux.derkeiler.com/Newsgroups/comp.os.linux.hardware/2005-09/0464.html>

---

**From:** kermi (cku192\_at\_yahoo.com)

**Date:** 09/21/05

Date: Wed, 21 Sep 2005 22:08:15 +0400

alazarev wrote:

> We use a popular consumer RAID enclosure device. It's 16 SATA drives,  
> with a built in RAID controller, hot swap everything, attaches to the  
> host via SCSI. We've been pretty happy with it up until a few weeks  
> ago. It is setup in in RAID 5. Nothing unusual about the setup. Host is  
> RHEL4-AS 64bit, filesystem is ext3.  
>  
> About a month ago, we saw some bad blocks on a drive, 5 of them in a  
> row. We ignored, we've seen it before and it's never been a problem. A  
> few weeks later, we got 4-5 more bad blocks. Did nothing. A few weeks  
> later, disaster, we got about 10 bad blocks in a row, and the last one  
> took out the filesystem. The host unmounted the filesystem  
> automatically.  
>

Well, you have been warned and did nothing. Nobody to blame ...

> The host logs showed that the disc containing the filesystem had a  
> failure:  
>  
> Sep 7 01:29:52 zeus kernel: attempt to access beyond end of device  
> Sep 7 01:29:52 zeus kernel: sdb1: rw=1, want=8072683984,  
> limit=2927171457  
>

Yes I have seen similar after hardware error; ext3 sometimes reacts very funny (I won't claim it is limited to ext3 only :) It was RHEL3 BTW.

[...]

> What angered us most was, there was our RAID system, sitting there  
> running, not even recognizing that it caused a massive host filesystem  
> failure. The RAID did see bad blocks, but it never marked a drive as  
> bad, never sensed any failure whatsoever.  
>  
> Now, my question is, isn't this problem exactly what RAID is supposed  
> to protect against? How could a RAID controller botch this up?  
>

[...]

- > *Before I start talking to other vendors, is this bad block issue a*
- > *problem for all RAID controllers? Shouldn't any decent RAID controller*
- > *be able to rebuild from parity whenever it senses a bad drive, or a bad*
- > *block? As soon as the RAID sensed a bad block, it should have known*
- > *there was data on it, and then either failed the drive, moved to*
- > *degraded, and then start reading from parity, so that the host would*
- > *never know about a drive failure? How could an enterprise level RAID*
- > *controller fail to do this?*
- >

You did not name manufacturer; not all of them are really enterprise level :)

But about general theory.

When RAID5 controller senses a bad block, it first should try to "remake" it. Sometimes writing over bad block fixes it; sometimes recomputing checksum and rewriting it fixes it; real enterprise level controller should log it as correctable error, ideally with extra qualifier to inform, that block was recovered without relocation.

If this was not possible, controller should use spare blocks to relocate bad one. Usually in this case contents (or parity) is lost and has to be rebuilt. The same (but on more massive scale) happens when the whole drive goes bad. In this case all blocks on drives are lost and their contents should be rebuilt on spare drive if available from remaining drives.

All is fine unless you have double fault. If during rebuild controller finds \*another\* failed block on the same stripe, data is irrecoverably lost. Apparently here is what happened, but it is hard to tell without more information.

Because during normal operation not all blocks in a stripe are necessarily touched it is quite possible that another drive has a bad block that you have no idea about \_until\_ controller must do rebuild. Enterprise level storage systems are using background verification that continuously runs and checks all blocks on all disks. It lessens double fault probability by discovering bad blocks early; still because it runs slowly (to lessen impact on performance) it takes much time so blocks may go bad after they had been verified as good :)

Another technique used by enterprise level systems recently is double parity (I am surprised that one quite known vendor still does not use it; apparently it believes that if you need that level of protection you use RAID10 and not RAID5 :). The idea is that you use two parity blocks per stripe instead of one – that gives you extra protection against double fault during rebuild. Most vendors support it as far as I know.

Recently I have read report about how probable double fault is. The values were quite amazing. For a SATA drives array of 5 disks has about 30% probability to encounter double fault over time. And you have 16 drives

you said ...

> *Any help or advice is appreciated, please feel free to email me too.*

>

To summarize:

- RAID5 is setup to protect against such types of fault
- RAID5 is able to protect against single fault only; it can't protect against double fault in a stripe
- double faults are (unfortunately) more and more likely now–a–days, especially with SATA disks

So when selecting vendor choose

- one with good logging. All errors (not fatal ones) must be logged; large number of correctable errors still indicate failing drive that should be considered for replacement
- one that provides for background verification. It helps to prevent such problems.
- one that supports double parity. That gives more protection against such faults (yes, it is more expensive because you spend two instead of 1 drive per RAID array)
- finally consider using RAID10 instead of RAID5.
- in any case storage controller must fail host IO *\*immediately\**; what storage controller must *\*never\** do is to fake success when it already has bad blocks. I suspect it may also be the reason for ext3 confusion.

=arvi=