

## Re: Raid 5 performance

**Source:** <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/performance/2004-02/0012.html>

---

**From:** Allen Landsidel (*all\_at\_biosys.net*)

**Date:** 02/07/04

Date: Sat, 07 Feb 2004 05:35:25 -0500

To: freebsd-performance@freebsd.org

At 04:39 2/7/2004, Uwe Doering wrote:

>Todd Lewis wrote:

>>I am using FreeBSD 4.9, with a 3ware RAID 5

>>I gig memory 2.8g p4

>>Three questions.

>>1. FreeBSD has a 16k block size. The RAID card is set at 64k

>> Block size(its sweet spot). My logic tells me that

>> increasing the block size to 64k would increase disk

>> read and write access. But, everything I read suggest

>> going above 64k is dangerous. Are there any recommendations

>> on performance a stability concerns when increasng the

>> block size to 64k when using a RAID controller.

>

>A RAID controller normally has nothing to do with the file system's block

>size. Are you sure that you're not mixing this up with the stripe

>size? Which stripe size to use with a RAID controller depends on your

>performance priorities.

>

>If there are a lot of concurrent disk operations a larger stripe size is

>better because then a single disk operation tends to be limited to only

>one disk drive, leaving the remaining drives free to perform other and

>possibly unrelated disk operations at the same time. On the other hand,

>if sequential i/o throughput is important a smaller stripe size is better.

I am compelled to step up here and say that this flies in the face not only of everything I have read, ever, about RAID -- but my own personal experiences as well on a variety of controllers and drives ranging from ATA highpoint/maxtor combos up to 4ch u160 ICP-Vortex/15krpm monsters.

My experience has told me that for mostly sequential I/O, bigger is better, up to a point. 128KB stripes are much faster than 16KB stripes on every combination I've ever used when it comes to sequential I/O.

Random I/O tends to prefer smaller stripe sizes because as you said, spreading things out over disks is better, and in random I/O the request sizes tend to be much smaller.

## freebsd-performance: Re: Raid 5 performance

My advice to our intrepid advice seeker is simply this. Get your system running under a load pattern as close to 'normal' as you can get. Then, run "systat -vmstat 1" and watch the output. Down with the disks is a row "KB/t" which is Kilobytes per transaction. Keep an eye on this number, and note the approximate value that seems to occur most of the time. Then when building your array, set the stripe size to the smallest value you can that is this size or larger.

This will ensure that the majority of your disk activity hits every disk, getting you maximum throughput. Set the value too low and every disk will be hit more than once for one request which means more controller overhead. Set it too high and you'll hit fewer (perhaps only one) disk on the majority of your requests, negating any performance benefit RAID could offer you.

If you plan to expand the array later by adding disks, take that into account and use a slightly smaller stripe size -- disks will be hit more than once until you add more, at which point you'll get closer to each disk only getting one request per transaction, which is the sweet spot for non-sequential I/O which is what most production servers (especially databases) do, most of the time.

I agree with one thing said here however -- the OS block size has little to nothing to do with this except for one thing -- you can be almost certain that your average request size will be equal to or some multiple of this value.

---

freebsd-performance@freebsd.org mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-performance>

To unsubscribe, send any mail to "freebsd-performance-unsubscribe@freebsd.org"