

# FreeBSD software RAID1: gvinum vs. gmirror

[Back in January](#) I introduced an upcoming series about FreeBSD software RAID implementations and how they compare to each other.

I know it's been some time since, but I was kept busy on some other stuff which prevented me in polishing up the results. My appologies for the delay.

Now, in this article I'm going to talk about FreeBSD's gvinum and gmirror for disk mirroring (aka RAID1).

The benchmark results shown below will illustrate the difference between these two implementations. To get a better idea of their real meaning, I'm providing you with the single disk results as well.

Now as far as software RAID goes in FreeBSD, gvinum (geom vinum) is not gmirror (geom mirror). They basically do the same, but in a slightly different manner.

gmirror does a direct approach in mirroring. It works with the low level device (the geom provider) directly to create the resulting mirror (the geom consumer). It is very specialized in this and supports also different balance algorithms which do have quiet an influence to performace as you will see later on.

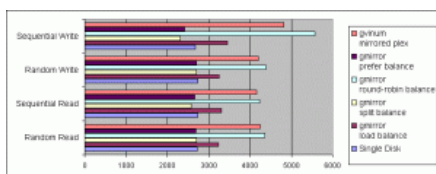
gvinum is all different. geom providers are referred to as subdisks or drives, which in turn are members of plexes, which again form volumes. In fact gvinum is a real volume manager that is capable of doing far more than simple disk mirroring. It supports also RAID0, JBOD and even RAID5. It is however not yet as mature and stable, especially when it comes to RAID5 - but this is a different story.

## # rawio benchmark results

rawio was choosen to measure I/O speed during concurrent access. rawio was set to run all tests (random read, seq read, random write, seq write) with eight processes on the /dev/mirror/\* and /dev/gvinum/\* devices.

Results for the single disk are provided as well to compare performance not only between the different frameworks but also against the native disk performance.

Click the images to see the actual result values and a chart.

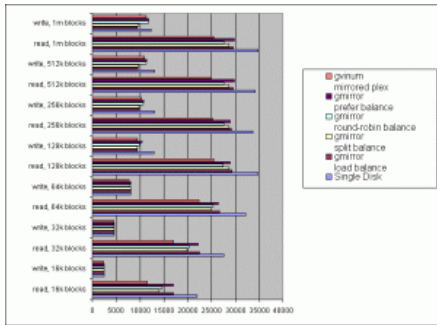


RAWIO	Single Disk	gmirror (load balance)	gmirror (round-robin balance)	gmirror (round-robin balance)	gmirror (load balance)	gmirror (load balance)	gvinum (mirrored plex)
Random Read	2719.9	2719.9	2719.9	2719.9	2719.9	2719.9	2719.9
Sequential Read	4700.1	4700.1	4700.1	4700.1	4700.1	4700.1	4700.1
Random Write	2719.9	2719.9	2719.9	2719.9	2719.9	2719.9	2719.9
Sequential Write	4852.2	4852.2	4852.2	4852.2	4852.2	4852.2	4852.2

## # dd benchmark results

dd was chosen to measure raw block access to /dev/mirror/\* and /dev/gvinum/\* devices. dd was set to run sequential read and write tests using block sizes from 16k to 1024k.

Click the images to see the actual result values and a chart.

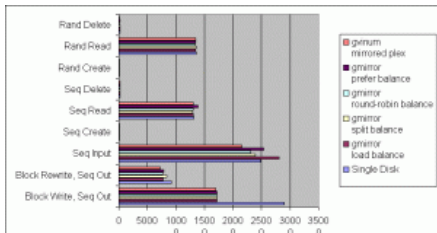


dd	Single Disk	gmirror	gmirror	gmirror	gmirror	gmirror	gmirror
		read balance	split balance	round robin balance	load balance	split balance	mirrored plex
read, 16k blocks	2196.43	1836.12	1362.13	1483.61	1609.66	1192.42	
write, 16k blocks	2472.36	2488.47	2474.01	2477.35	2477.38	2426.25	
read, 32k blocks	2121.57	2228.37	1625.43	2025.19	2229.27	1734.93	
write, 32k blocks	4033.53	4252.48	4052.23	4033.27	4254.52	4433.25	
read, 64k blocks	2012.44	2564.77	1456.07	2030.74	2554.11	2043.33	
write, 64k blocks	6037.57	6017.73	6034.26	6032.01	6030.66	7711.00	
read, 128k blocks	34834.72	25265.95	2872.99	27483.33	26983.25	26481.00	
write, 128k blocks	12642.38	14782.22	1395.54	1654.96	13423.52	1623.17	
read, 256k blocks	33637.44	26130.18	36220.23	21933.73	26236.10	25182.52	
write, 256k blocks	12693.34	18027.74	3871.44	10640.68	10896.76	10296.42	
read, 512k blocks	34139.23	29514.03	20726.41	22529.65	29775.02	24891.66	
write, 512k blocks	12995.98	16254.91	3101.03	11295.33	11396.57	10801.53	
read, 1m blocks	34793.08	28506.15	20742.91	27699.99	29322.02	25436.35	
write, 1m blocks	12624.49	14275.28	3640.74	11807.54	11803.44	11260.20	

### # bonnie++ benchmark results

finally, bonnie++ was used get pure file system performance.

Click the images to see the actual result values and a chart.



bonnie++	Single Disk	gmirror	gmirror	gmirror	gmirror	gmirror	gmirror
	Read	Read	Write	Read	Write	Read	Write
	balance	balance	balance	balance	balance	balance	balance
Block Write, Seq Op	28624	17148	17148	17148	17148	17148	17148
Block Read, Seq Op	3020	7723	8286	7618	7841	7283	7283
Seq Read	24619	26111	25875	25371	25383	21563	21563
Seq Write	38	32	32	32	32	32	32
Seq Read	13156	13859	13819	13813	13811	13008	13008
Seq Delete	168	168	162	168	162	164	164
Read Create	88	74	72	73	74	71	71
Read Read	13543	13504	13549	13561	13512	13419	13419
Read Delete	192	182	182	183	182	182	182

## # conclusion

Looking at raw disk access I must conclude that none of the frameworks beats single disk performance in overall when it comes to blockwise input with dd. One get's however quiet an impression about the overhead imposed by both gvinum and gmirror.

Now since 'dd' is very synthetic by it's nature, rawio is much better to see how the devices would perform under a more "real-life" situation.

Winners are both gvinum (mirrored plex) and gmirror (round-robin balance algorithm), while the later is still a bit faster though. Although rawio benchmark results may look low, these numbers where achieved by running 8 processes at once. They'll reflect best what could be expected in a true multi-user environment with concurrent access.

Finally for bonnie++ we see some interesting results. Performance is almost identical for all implementations expect for block write access, where the single disk is the unbeaten winner.

One must keep in mind that the first six tests performed by bonnie++ (rand delete/read/create, seq delete/read/create) are limited by I/O performance of both the system bus and the device itself. The hardware I used for testing was capable of about 160 - 170 I/Os per second. I admit that results could be different if the tests are re-run on decent hardware with a higher I/O throughput. It's possible that modern hardware reveals an I/O barrier for abstracted devices which cannot be seen from my tests.

Personally I prefer using gmirror over gvinum because of it's more simplistic configuration approach. In terms of performance, gvinum is almost as good as gmirror (round-robin algorithm) but still can't yet keep up with it.

While both may look slower altogether in synthetic tests (dd and bonnie++) than the other balancing algorithms, rawio reveals their potential in a manner which is comparable to real field duty.

The next article will discuss gvinum and gstripe for RAID0.