FreeBSD software RAID0: gvinum vs. gstripe

Back some time I announced reviewing FreeBSD's geom software RAID implementations.

Todays article compares geom stripe (gstripe) along with geom gvinum (gvinum) for disk striping (RAID0).

All testing was done on the same hardware as before to get results comparable to previous tests.

Benchmarks were taken using stripe sizes of 64k, 128k and 256k and measured using dd, bonnie++ and rawio as before.

As for the technology gstripe follows the same approach than gmirror which I look at previously.

# 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/stripe/* 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.

529

532

# dd benchmark results

dd was choosen 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.

523

520

# bonnie++ benchmark results

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

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

517

513

# 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/output with dd.

gvinum generally performs better than gstripe except when using 256k stripe sizes.

Now since â??ddâ?? is very synthetic by itâ??s nature, rawio is much better to see how the devices would perform under a more â??real-lifeâ?• situation.

Although rawio benchmark results may look low, these numbers where achieved by running 8 processes at once. Theyâ??ll reflect best what could be expected in a true multi-user environment with concurrent access.

As from the results there is no absolute winner, as depending on the stripe sizes either of both implementations out-performs the other.

Finally for bonnie++ we see some interesting results. Performance is almost identical for all implementations.

One notable exception was seen with gvinum (64k stripe size) which clearly outperformed its competitors..

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. Itâ??s possible that modern hardware reveals an I/O barrier for abstracted devices which cannot be seen from my tests.

Personally I prefer using gstripe over gvinum because of itâ??s more simplistic configuration approach. In terms of performance, gvinum seems to offer superiour performance when it comes to disk striping.

The next article will discuss gvinum and gstripe for RAID10.