FreeBSD software RAID comparison: Introduction

FreeBSD comes with support for multiple software RAID implementations to date.

ata, which is not really a RAID toolkit but the ata device driver itself.Nevertheless it provides sort of RAID functionality by also interfacing to popular low-end RAID controllers, sometimes also referred to as Fake-RAID controller (highpoint, promise and alike).
It has support for enabling RAID on non-RAID capable ata controllers, too.ata is IMHO the worst RAID solution out there, and as such should be avoided at any cost. Please read as I'm writing: I'm not telling you the ata driver is bad, but I think it shouldn't tamper with RAID.
RAID should never be implemented by means of a device driver which does some scary magic inside. A real RAID controller (hardware RAID) or a volume manager (software RAID) should be used instead.

- ccd, the concatenated disk driver, used to be a standalone framework in previous FreeBSD versions. It has been integrated with geom (geom_ccd) in FreeBSD 5 and IMHO seems only to exist for historical and backward compatibility reasons by today.
- vinum was also a standalone RAID framework in previous FreeBSD versions. It too was merged into geom (geom_vinum) in FreeBSD 5.
- geom, which is basically an I/O abstraction layer which serves as stackable framework to multiple subsystems like vinum volume manager (which does jbod, raid 0, raid 1, raid 5 and combinations thereof), mirror (which does raid1), stripe (raid0), and many others.geom provides both functionality and reliability in a proper implementation. I think it might outrun big players like Veritas Volume Manager one day.

So what I'm going to do is a performance comparison (benchmark) of the different geom classes providing RAID1 (mirroring), RAID0 (striping) and RAID10 functionality.

I will neither review ata's RAID capability nor the geom_ccd class for the obvious reaons mentioned before.

All tests have been taken on a Sun E450 with 4 ultraSparc-II CPUs @ 400 MHz. It has 1 GB RAM aside and 6 disks (SEAGATE ST318404LSUN18G) interfaced to the Symbios Logic scsi host bus.

To get accurate results, all tests were run five times, takeing the average values from all tests.

Testing was done using three methods.

- rawio was used to get low level I/O performance for the geom consumer under a concurrent load simulation (multiple tasks reading/writing in parallel).
- dd was used to get low level I/O performance for the geom consumer in sequential read and write (single task).Sequential read was tested by writing 256 MB of data from the consumer to /dev/null useing block sizes from 16k to 1m. /dev/null was used as output device to circument I/O bound rate limits while reading from the provider.Sequential write was tested by writing 256 MB of data from /dev/zero to the consumer useing block sizes from 16k to 1m. /dev/zero was used as input device to circument I/O bound rate limits while writing to the provider.
- bonnie++ was used to get pure filesystem performance by writing/reading 1 GB files.

I must admit that these tests are truely synthetical and do not reflect actual performance to be expected in a production grade environment. But this was not the idea of these tests anyway.
What I wanted to see was if there is a difference in the geom classes in terms of their pure performance and the overhead the impose, basically if there is a "winner" and a "looser" amongst the different implementations.

You may want to check out my upcoming articles on these topics to learn more:

FreeBSD software RAID comparison: gvinum RAID1 vs. gmirror
FreeBSD software RAID comparison: gvinum RAID0 vs. gstripe
FreeBSD software RAID comparison: gvinum RAID10 vs. gmirror+gstripe
FreeBSD software RAID comparison: gvinum RAID5

And don't forget: the results shown later on reflect the performance on my particular hardware. There is no guarantee to get the same results on different hardware. It is very likely though that the results found here will be similar comparing the overall performance and the overhead ratio on other platforms.