What I Learned This Month: DS8300 to DS8870 Migration

Scott Chapman

American Electric Power

This has been another busy month of hardware migrations. In this case, we've migrated our disk subsystem from a DS8300 to a DS8870. I had some performance concerns about this because of the architectural changes that we were making. So far it looks like most of my concerns were unfounded.

I was not as concerned about this hardware change as I was the z10 to zEC12 migration (see last month's article). But I did have concerns because we were making some significant architectural changes in how the subsystem was configured. We had used RAID-10 based subsystems for at least the last 3 generations of storage, but we were moving to a RAID-5 based implementation for this one, primarily due to cost and capacity concerns.

In case you're not familiar with the terminology, RAID-10 represents a data protection scheme that utilizes striping and mirroring. In other words, the "chunk"[1] of data to be written is broken up and written in parallel across multiple disks. That process is mirrored across another set of disks. If one disk fails, the data is simply read from its mirror copy.

In a RAID-5 scheme, the data is still broken up across multiple disks, but instead of mirroring the data on a second set of disks, parity is calculated from the first set of disks and written to another disk. If one disk fails, the missing piece of data can be recalculated from the stored parity data.

There is more overhead for writes to RAID-5 storage because the parity must be calculated. Although the calculation itself is fairly simple, in some cases (if the write is for a partial "chunk") it may even be necessary to read the non-updated part of the chunk to correctly recalculate parity. In some implementations, RAID-10 might have an advantage in read performance as well because either copy may be read. Similarly, in the case of a disk failure, the rebuild process is simpler for RAID-10: just make a new copy of the mirror of the disk that failed. In a RAID-5 environment, the data from all the other disks in the RAID set has to be read to recalculate the data that was lost.

[1] This is an intentionally vague term as the exact implementation details vary between storage subsystems.

In short, simpler is often better, and fundamentally, data protection schemes that rely on mirroring are simpler than those that rely on parity calculations.

The downside is the storage overhead: if you mirror, you lose half of your total capacity to data protection. Parity-based schemes may only need 20% or so of the capacity to store the parity information. While storage costs are in a continual decline, premium enterprise-class storage still represents a non-trivial expense.

The other thing I was concerned about was moving to larger drives. We were approximately doubling the size of the individual drives. Again, this is a good thing from a cost/GB perspective, but from a performance perspective, fewer spinning disks means fewer concurrent I/O’s on those back-end devices, which means greater chance for queuing in the storage subsystem waiting for I/O to or from the actual spinning disks.

The vendor had modeling data that said that our average I/O response time would actually go down very slightly with the new subsystem that had larger drives and RAID-5. I had no real argument with that: we were also adding more cache and zHPF and we already were getting pretty good cache hit rates. All writes go through the cache, and so the back-end storage (usually) has limited impact on the subsystem’s write rates. Reads that come from the cache are not dependent on the performance of the drives. So if you add more cache and improve transfer times with zHPF, then the overall average performance will likely improve.

But my concern was not the well-performing I/Os that come from the cache, but rather the cache misses that are dependent on the back-end drive performance. Also, out of all the I/Os that are executed in the system, some are really asynchronous I/Os that are not holding up the continuing execution of an application. The I/Os that really matter are the long ones (cache misses) that something important is waiting on. Total average response times are interesting, but not necessarily entirely representative of the impact to the business work.

It seemed to me that moving to RAID-5 with larger drives would likely cause an elongation in cache miss I/O elapsed times. However, if you reduce the number of cache misses by adding cache, then the elongation of the smaller number of cache misses may not matter in total.

As part of our effort to reduce the performance risk of both the processor and the DASD changes, we also added SSD drives to the subsystem. While SSDs are still fairly expensive, the cost for enough SSD storage to hold our largest production database was not unreasonable.

So how did it work out? The implementation was only very recently completed, but the new configuration seems to be working well. As the vendor's model suggested, our overall average response times did improve somewhat. Of course when you're already running in the low single digit millisecond response time range, there's only so much improvement that can be made. The minimum response times I'm seeing have decreased, likely due to faster channels and faster processors in the storage subsystem. Our periodic brief response time spikes[2] for individual volumes also seem to be lower.

RMF reports statistics on the back-end arrays as well. For the arrays with hard drives (not SSD), those show that the average read response time to the actual physical arrays has declined slightly, while the write times have increased on average. However, I was surprised to see that the write response times were much more consistent and don't spike as high as they used to. The SSD arrays are noticeably faster than the hard drive arrays, especially for write activity. My guess is that the writes perform particularly well on SSDs because of the reduction in time it takes to re-read the existing data to recalculate the parity. But since all writes are cached, that improved physical write response time makes relatively little difference in the total response time visible to the host.

It's also interesting to me that RMF reports average read response times on the physical arrays of about 2ms, which is pretty much what the average rotational delay for a 15K RPM drive calculates out to. Given that there's seek and transfer time as well, that seems to be a bit unlikely to me. So I'm not entirely sure at this point what to make of the value, but on the old box it was typically over 3ms. The RMF report puts SSD read times at around 0.7ms, which certainly seem plausible given that there's no rotational delay or seek time.

Due to various reasons, the back-end configuration of the old DASD was not perfectly balanced. When we configured the storage on the new box, we took some extra effort to rebalance the SMS storage pools across the physical back-end RAID arrays. I believe that some of the reduction in variability that we've seen is due to this rebalancing of the physical resources. I also created a little reporting tool that should help us keep a better focus on how the volumes for a given SMS storage pool are allocated to the physical resources. I'm hoping that will help us keep things more balanced going forward.

Practically, all of those response time metrics are more or less meaningless. What really matters is the impact to the actual business work. In that regards, online response time did improve slightly, but batch work did not noticeably change. Given that we don't yet have everything migrated to the SSDs that will eventually be going there, there may still be some opportunity for improvement.

[2] In the past we've attributed these to the DR replication process. However, I'm again wondering if that's the entire explanation for the average response times for a volume spiking to tens of ms over a 100 second interval; especially since it's not widespread at any given point in time and since it happens on volumes backed by SSDs as well.

So it appears that my fears regarding RAID-5 in the DS8870 were unfounded, at least in our configuration. From the z/OS perspective, overall average response time improved, although I can only speculate on how much of that was due to each of the several configuration changes (faster channels, zHPF, more cache, SSD, faster storage processors). RMF reports several technical measures improved, some to levels that seem almost impossibly good. While I'm interested in understanding the technical measures, the real point is that the upgrade had a net positive impact as measured at the level that's important to the business.

As always, if you have questions or comments, you can reach me via email at sachapman@aep.com.