Q & A From Hitachi Data Systems WebTech Presentation:
“RAID Concepts”

1. **Is the chunk size the same for all Hitachi Data Systems storage systems, i.e., Adaptable Modular Systems, Network Storage Controller, and Universal Storage Platform, or are they different?**

For the Adaptable Modular Storage series, the chunk size defaults to 64KB, but can be adjusted to 256KB or 512KB. For the Universal Storage Platforms, the chunk size is fixed at 512KB for Open Systems attachment.

2. **Why do we measure random I/O in IOPS, but sequential I/O in MB/s?**

For random I/O, each I/O operation handles a small block such as 4KB or 8KB.

For random I/O on the port processor, since the blocks are small, it doesn't really matter how big they are, since nearly all the microprocessor utilization (% busy) is associated with starting up / shutting down the I/O operation, and very little MP Busy is for tending data flow in mid-transfer. So the MP Busy doesn't vary as a function of block size.

Similarly, for disk I/O on the parity group, random I/O block sizes are so small that the data transfer time for the block is less than 1% of the total time it takes to perform a disk I/O operation. So both at the port level and the disk drive level, for random I/O all we really care about is how many I/O operations per second we handle.

For sequential I/O operations, the block sizes are designed to be long, being aware that we deliberately want to make the block sizes big enough so that we get a lot of data moved for the overhead associated with each I/O operation.

At the host port level, for sequential I/O operations, the block size is designed to be big enough so that the Fibre Channel path’s data transfer speed (2Gbit/sec or 4Gbit/sec) is reached before the microprocessor gets too busy – so for the port, all that matters is MB/s, not IOPS.

Then as we discussed during the RAID Concepts presentation, for sequential I/O operations, the host to subsystem block size is quite independent of the block size that the subsystem writes the data to disk with, as cache acts like a holding tank to accumulate enough data to write full stripes to disk. The chunk size on disk, which is the unit of transfer to each disk to stage / destage sequential data, is selected to be substantial so that we transfer “lots” of data for each I/O operation. The chunk size of 512K used by the Universal Storage Platform family (and selectable for the Adaptable Modular Storage family) represents roughly one disk rotation worth of data transfer time. This compares to over one disk revolution for mechanical positioning before data transfer starts. Thus whereas for random I/O, data transfer represents roughly one percent of the total disk drive busy time, for sequential I/O, data transfer represents roughly one half of the total disk drive busy time.

This 512KB block size (chunk size) represents a good compromise for two reasons:
1) Intermix between random and sequential I/O on the same parity group. The longer the sequential block size, the longer that random read miss I/O operations have to wait in case the subsystem happens to be (asynchronously) doing a sequential stage or destage on the same parity group at the time the random read miss arrives.

2) The bigger you make the chunk size, the larger the sequential stripe becomes, and the more often you will encounter the situation where you only receive from the host during a series of write operations enough data to partially but not completely fill a stripe before the data has been hanging around long enough so that you just have to write it to disk.

This is a significant factor, and happens often enough to significantly affect sequential throughput, as when you only have a partial stripe to destage, you must perform the destage the same way you destage random I/O, which for RAID-5 and RAID-6 means you need to read old data and read old parity before you can write the new data and parity.

This extra I/O workload to randomly destage partial stripes can significantly impact the throughput you expect to get with RAID-5 and RAID-6 sequential writes.

For the Adaptable Modular Storage series, cache sizes are smaller than the Universal Storage Platform as well, and there needs to be enough space in cache to accumulate full stripes for all sequential streams you have in progress at one time in order for sequential writes to flow at “full stripe write” speed.

The stripe size is also a factor of how many drives there are in the parity group, and in order to keep sequential writes operating in “full stripe write” mode, it’s a good idea to avoid the larger RAID-5 / RAID-6 parity group sizes that are possible on the Adaptable Modular Storage.

Avoiding larger RAID-5 / RAID-6 parity group sizes on the Adaptable Modular Storage also reduces the performance impact of drive failure. With no failing drive, to read a random record involves a single I/O operation to a single drive. If that drive has failed, the data can be reconstructed, but to do that requires reading from all of the remaining good drives. The number of I/O operations to do the reconstruction is thus proportional to the size of the parity group.

3. Is it possible to control what part (outer, mid, inner, etc.) of the disk drive that data is kept? So higher usage requirements, file systems, can be placed in a better performing area of the disk.

The only thing that is faster at the outer diameter of the drive is the data transfer rate. (Blocks are numbered starting at the outer edge, so the data transfer rate starts out fastest at the beginning of the drive, and reduces in a series of steps to the innermost tracks at the end of the drive.)

So because for random I/O, the time to transfer a block is on the order of 1% of the time it takes to do an I/O operation, the difference in data transfer rate really doesn’t matter.

For sequential I/O, yes, in theory the higher data transfer rate should be significant. But firstly, sequential staging / destaging is an asynchronous process, so the difference in transfer rate will only affect overall parity group busy. Next, sequential stages / destages are done in parallel across the parity group, whereas all the data concerned is funneled onto the one host port, so the sequential throughput rate of the parity group is usually not the limiting factor.

Only if you are truly moving vast quantities of data sequentially through a parity group where the Fibre Channel port stays saturated at max throughput (400MB/s for a 4Gbit/s port) do you really care about the sequential throughput of the parity group, and in that case having the data at the outer edge will help. In this case, you can also configure larger parity groups for the Adaptable Modular Storage (assuming you only have a few things happening at the same time so you won’t over commit cache capacity), and in the case of the Universal Storage Platform family you can configure “concatenated parity groups” where two or four 7+1 parity groups can be striped together.
4. Do Hitachi Data Systems products use RAID 1+0 or RAID 0+1?

People like to discuss the difference between a mirror of stripes versus a stripe of mirrors, but when this all happens within one parity group the only difference is the numbering of the drives within the parity group.

The Universal Storage Platform family does this in a 2+2 as `data_1 data_2 copy_of_data_1 copy_of_data_2`, so it’s a mirror of stripes.

5. Is there a published list of access density recommendations for optimal performance per disk type?

I’m sorry, we don’t have a list of which host workload access densities are recommended for which drive types and RAID levels. Have your Hitachi Data Systems representative contact me within Hitachi Data Systems to discuss further.

6. How does the parity bit generation affect CPU utilization?

In the Universal Storage Platform family, parity generation does not affect front end or back end microprocessor utilization, as there are dedicated microprocessors to handle this task. I don’t know for sure, but for the Adaptable Modular Storage family, I imagine calculating parity contributes to overall microprocessor utilization levels.

7. Say you have Oracle doing 8K random writes. If you have a 4P+1 RAID-5 with 2K segments you wouldn’t have any write penalty.

Sorry, no. Random write block sizes are much smaller than the size of one disk chunk. Thus all 8K goes to a single chunk on the disk. The only time that you would not have a write penalty is if the entire stripe is written from the host. For a 4+1 RAID-5 parity group, the default chunk size is 64K, and thus to have no write penalty, you would need to write 4x64K = 256KB and this write would need to be aligned on a 256KB boundary within the array group. (In other words, the same as a sequential write destage.)

8. Is the impact of RAID so big for random writes? All I/Os to the physical disks are in parallel. For each host I/O there is only one read pass and one write pass. Correct?

Yes, there is only one read pass and one write pass. However, it is really only the total number of I/O operations that we care about, since all array group I/O operations (other than read misses) are asynchronous to the host. For a random write destage that takes 4 I/O operations, yes, these take place in two passes. In the first pass we read old data from one drive and old parity from the other drive in parallel. Then in the second pass we write the new data and the new parity in parallel. But these I/O operations all take place asynchronously after the host write operation has completed. We care about the number of I/O operations because this is what determines parity group busy. If the parity group gets too busy then 1) read miss service time will degrade, and 2) pending write data will build up in cache. Ultimately, if the parity group stays too busy for too long, cache will fill with pending write data and the subsystem will have to make host I/O wait until this condition can be cleared by destaging enough pending write data to disk.

9. Is it necessary to stripe at file system level (LVM) since we have done RAID at hardware level? Will there be significant performance improvement combining hardware RAID and host level LVM striping?

This is a question for which I don’t have a clear answer. Striping in general whether at the host level or the subsystem drive level spreads activity across multiple disk drives. This is a good thing if you have one volume with very high activity level. This is good to spread activity around so that you don’t have “hot spots”. But it’s not a good thing if you want to isolate the performance of one application from another, because usually when we do striping there are other logical volumes striped across the same physical disk.
drives and thus striping makes it more likely that activity on one logical volume will impact the performance of the other logical volumes striped across the same physical drives.

10. Generally for database OLTP apps: should OS cache file and data base redo/transaction logs go on RAID 0+1 and data base main file on RAID 5?

Generally speaking, logs involve sequential I/O. Sequential I/O in general performs better than random I/O on RAID-5. So the tendency is to put logs on RAID-5 at activity levels (access densities) where other more random workloads will have been put on RAID-1. Having said that, sequential I/O on RAID-5 only works ideally when the volume of writes is big enough for the subsystem to accumulate full stripes within a short interval (measured in seconds). Otherwise, even if the I/O pattern is sequential, if full stripes don’t accumulate, the data will still need to be destaged using the random destage method.

So to answer your question, I would put the main database file on RAID-1 and the logs on RAID-5.

11. Do you have comments on SAS drives compared to Fibre Channel and SATA?

SAS (Serial Attach SCSI) is a newer interface that may replace Fibre Channel over time. There is no particular performance difference between SAS and FC, other than the link speed in Gb/s. Having a higher link speed on the host to subsystem link (which now is all FC) all by itself without increasing microprocessor speeds just lets you run higher MB/s.

On the back end, where the drives are attached to the subsystem, having a faster link data rate allows more drives to share a single port, but does not improve the performance of I/O operations to a single disk drive. That is because the interface is already faster than the drive data transfer rate. If you are already waiting for the drive, then making the interface faster doesn’t help when waiting for that one drive. However, if the drive interface data rate is increased, then the drives get on and off the interface faster, thus allowing more drives to use the same interface before the interface reaches its performance limit.

There are 3 drive interfaces in common use today, Fibre Channel, SAS, and SATA. Generally speaking, Fibre Channel and SAS are used for high performance expensive drives, and SATA is used for low performance large capacity drives. But the cost of the drive and the performance of the drive are not influenced by the type of interface. The cost comes from the number of heads and media needed to achieve the capacity – this is determined by the RPM (revolutions per minute), as at higher RPM’s the platters need to be smaller and thus you need more platters, and also when we make the RPM faster, people expect the seek time to be faster too, so the actuator motor that moves the arm back and forth needs to be much bigger and faster and more expensive, too. But the type of interface really doesn’t make any difference. In the past, the ancestors of SATA drives were lacking some features such as tagged command queuing, but nowadays all SATA drives have TCQ.

Thus to you as the customer of the subsystem, it really doesn’t matter what drive interface is used in the back end of the subsystem – it’s an implementation detail.

Over time we may see SAS replace Fibre Channel just because SAS and SATA share the same physical hardware (only the protocol is different). This means that SAS and SATA will be able to share the same connection infrastructure. SATA only supports single porting and point to point connections, but there’s a way to encapsulate the SATA protocol over a SAS connection, so in future it may become more common to see the ability to configure either SAS or SATA drives on the same connection infrastructure.

But don’t make the mistake to say that SATA is cheap and SAS/Fibre Channel is expensive/fast. The speed and cost come from the drive behind the interface, not the interface itself. (OK, there are some tiny differences, but they are not significant compared to the cost of heads / media / spindle motor / actuator.

12. How do we calculate I/O density of the applications, what tools do we use to measure contention in the drives. How do we demonstrate latency for I/O density factor?
The I/O access density for an application is the I/O rate divided by the storage capacity, and it’s expressed in IOPS per GB.

The contention for disk drives is measured in disk drive utilization (% busy). Performance measurement software such as Hitachi HiCommand® Tuning Manager will display this.

Latency is the term used to describe how long you have to wait in a disk drive for the data to come under the head once you have completed moving the access arm to the desired position (seeking). For single standalone I/O operations, the average latency is ½ a turn. When the drive is heavily loaded and within the drive there is a queue of I/O operations, the drive will look at the destinations for all the queued I/O operations and will perform the I/O operations in such a sequence as to minimize the mechanical positioning delay, both seeking and latency. This feature is called “tagged command queuing” (TCQ). Thus when a drive gets heavily loaded, the average latency will go down below ½ a turn, and the average seek time will also decrease. There is no way to measure or report on these reductions other than to notice that the drive’s throughput rate is higher than you would expect. But please note that although TCQ improves throughput, it does so at the expense of making response time worse. Because the drive reorders the execution time, it sacrifices the wait time for individual I/O operations for the greater good. You can think of having the post office delivery person going to every house on their route in the order that letters arrive at the post office vs. going to the houses in the order of the houses on the street. With a disk drive the performance difference is not as extreme as this example, but the principle is the same.

With TCQ, you can expect to see improvements as much as 30% to overall throughput, but at very high disk drive busies. TCQ only improves throughput when the I/O operations are independent, meaning that they can be executed in any sequence.

13. What is meant by “density rates of data”?

Access density measures the intensity of the access to data. It’s a characteristic of the workload no matter what type of disk drive it’s put on. Access density is computed as the total IOPS divided by the capacity, and is expressed in units of IOPS per GB.

14. What combination of perfmon counters are best for determining random I/O?

Specific settings for particular performance measurement software are beyond the scope of this talk, but for random I/O workloads the most important specs are 1) access density and 2) read:write ratio.

15. Does Ian host a blog?

Sorry, no.

16. Why does the head of a LUSE generate more I/O than LUSE members?

The logical block addresses within a LUSE (or any logical volume) are mapped from the beginning to the end. If the first LDEV within a LUSE (the head LDEV) is receiving more I/O operations than the other LDEVs within a LUSE, this is just because the host application is issuing more I/Os to the address range near the beginning of the logical volume.

If you configured a LUSE in order to spread the I/O load over more than one parity group, you might want to look at using “concatenated parity groups” instead. With concatenated parity groups, each stripe goes across all the parity groups, and thus this does a much better job of distributing random I/O (which tends to be clustered in space – locality of reference) than using a LUSE. For the same reason, concatenated parity groups can give you higher sequential throughput if you have an application that steadily drives a port at 400 MB/s to a single LDEV.

17. What factors determine access density needs?

Access density is a characteristic of the workload, and therefore isn’t something that you can adjust. You might see access density vary simply because the entire logical volume might not be filled, and thus if the
files are moved from one logical volume to a bigger or smaller volume, the access density that you see at the level of the logical volume (LDEV) may have changed, but really the number of I/O operations that go to one file do not vary depending on where the file is placed – unless, of course, that the storage the file is on can’t keep up with the workload.

18. If you do heavy video editing where you simply want to take video clips from various sources and put them together into a new video stream (using Adobe’s suite), would RAID 1 be better or RAID 5 on stand-alone workstations where disks are local to the PC?

Good question – I don’t know. I would assume that video would be something where sequential data rates would be high enough to make RAID-5 work well, but with today’s drive prices, why not just use RAID-1?

19. RAID 6 and 300GB drives: recommended or not?

RAID-6 is the only RAID level that protects against two drive failures in the same parity group. However, RAID-6 also has the heaviest RAID performance penalty, and thus the access density that RAID-6 can handle on the same drives is lower than the other RAID levels.

Use RAID-6 where data integrity is the most critical, spread the data out over enough drives to keep the drive busy under 50% busy.

20. What is going to happen with bigger size drives? RAID 6 for 750GB drives and move RAID 5 to 300 10K?

The bigger the drive, the lower the intensity of the access to the data that can be sustained. If you fill a 750GB drive up with data, that data had better be VERY lightly accessed, or else the drive won’t be able to keep up with the I/O rate.

Ask your Hitachi Data Systems representative to help you figure out what the most cost effective combination of drive RPM, drive capacity, and RAID level is for your application. Have your Hitachi Data Systems representative contact me if they need help.