Queue depth calculation

Most Hitachi disk subsystems have similar allocations and limits. Consult the User's Guide for each model and host OS for specifics. The purpose of this solution is to explain the general design criteria and limitations.

512 / #LUNs / #Hosts ⇐ 32 is what you will find in most of the manuals.

In English: ((512 divided by number of LUNs) divided by number of hosts)) less than or equal to 32.

Note: Set queue depth equal for all hosts on the port without regard to host groups.

Note: There are 512 or 256 queue slots (model dependent). There is no problem with queue depth values higher than calculated above unless more than 512/256 total simultaneous commands to the port are exceeded or 32 commands are exceeded for a LUN. The formula above guarantees you will never exceed the queue capacity. The maximum performance may be achieved at higher queue depth values. The value above is quite general and assumes all LUNs are online and available to all hosts and that they are all active.

We commonly make this queue business harder than it has to be. The value of 256 or 512 is simply how many slots are physically allocated for the port or host group. The ucode will not take more than 32 from that pool for a given LUN. The design assumes many LUNs will be assigned to the ports and that the machine will be highly utilized. Huge LUSE LUNs tend to circumvent the design and may not perform well at QD=32.

If a machine is fast enough and can get the work out, the number of queue slots actually used is small. The book values simply guarantees the host can never overrun the possible slot allocations in the subsystem and present queue full back to the host. That of course does not mean it will be the best performer. The number is a compromise and based on an assumed LUN count.

A very low LUN count is not the best match for the design queue limit. In other words, 6 10TB LUSE LUNs will not perform the same as 60 TB of separate LUNs. If you compensate for the difference in design expectation and the true LUN scheme by using >32 QD, you are outside the officially supported range.

High QD settings may be necessary to allow the host to issue enough I/O to perform adequately but it would be better to reconsider the number of LUNs being used. If the disk subsystem is fast enough, the actual queue slot allocation is irrelevant since little may be queued anyway. This can only be said of very unusual configurations with a small number of huge LUNs. (not recommended)

If you understand the design criteria, you may be able to operate more efficiently at higher than “book” QD but you forgo the “guarantee” the host can never bump the queue full condition. Sometimes we are artificially throttling the host to a specification that expects to service hundreds of LUNs instead of a few.

Setting the queue depth at a high value may allow the host to perform better simply because the disk subsystem is underutilized or not optimally configured. If an actual queue full condition is arises, there is risk of the server going down and data integrity issues based on how the host handles the retry or recovery.

If QD is set higher than “book” outside the bounds specified in the User's Guide, the “guarantee” that the host can never overrun the disk subsystem is null and void but you could possibly increase your throughput. If the disk subsystem is configured with a small number of huge LUSE LUNs, the host may be artifically throttled at QD=32. If possible, use a host based product to create concatonated LUNs instead of LUSE then all this discussion of operating outside recommended values is moot.

Caveat: If you use QD values outside those specified in the user's guide, it may provide good results but you are responsible if it creates trouble. You may be asked to correct QD to book value if you open a case with tech support and they feel it could be related to queue depth. Most applications will do very well with normal QD values. It is not expected nor recommended that the user create multi-terabyte LUNs via LUSE and expect great performance.

Note: TRC does not endorse or support QD values outside the range specified in the User's Guide.

hds, qd, howto