Essentially, the shift parameter increase causes a ^2 increase in the number of keys available to associate with the deduplicated and compressed chunks of data. In turn this allows for a greater size of addressable physical storage.

|  |  |  |
| --- | --- | --- |
| Dictionary Size in GB and size on disk | Max addressable physical storage space in TB for the 32KB average deduplication chunks with 0% compression savings | Max addressable physical storage space in TB for the 32KB average deduplication chunks with 50% compression savings (16K chunks written to disk) |
| 64 (cloud optimized install) | 90.6 | 43.5 |
| 128 | 181.3 | 90.6 |
| 256 (standard and large install) | 362.6 | 181.3 |
| 512 (384GB on disk) | 725.3 | 362.6 |
| 1024 (640GB on disk) | 1450.6 | 725.3 |
| 2048 (1152GB on disk) | 2901.3 | 1450.6 |

The trade-off to increasing the shift number is an increased size in the dictionary, held as part of the QoreStor installation.

There is also a memory increase requirement (depending on what the QoreStor server already has) to allow the new dictionary size to function correctly. The makeup of required resources is, on-disk size, cache memory required, and data structure memory needed. There are no extra CPU requirements when increasing the size of a dictionary.

For example:

If a physical addressable storage size of 1PB (mix of local and cloud) is required, then depending on the average chunk size, post dedupe and compression, the following would apply...

|  |  |
| --- | --- |
| No of keys to support | avg chunk size post dedupe+compression |
| 137,438,953,472  (138 B) | 8k |
| 68,719,476,736    (69 B) | 16k |
| 34,359,738,368    (35 B) | 32k |
| 22,906,492,245 (23 B) | 48k |

If the incoming data deduplicates and compresses extremely well you can end up with more 8k and 16k chunks, thus not filling so much physical storage, allowing more data to be stored. The knock-on impact is that if the physical storage is filled to its maximum, for example, with 8k and 16k chunk sizes the dictionary would need more keys to reference a higher quantity of chunks. The better the deduplication and compression, the more keys you may need.

It's hard to assess what the deduplication and compression outcome may be as it's dependent on data types, change rates and number of data copies held.

We know the number of data copies to be held over time as that is set in the retention policy from the backup software. Change rate is harder to work out, but we can add a variable in to include it in any calculations. The change rate is what represents unique data being ingested when compared to existing deduplicated data.

The relationship between memory & disk for dictionary pages.

