

Don't be duped by inflated data reduction ratios and guarantees





Data Reduction Traps

Data reduction, as the name indicates, involves reducing the data footprint while transmitting or storing information. Pattern removal, data deduplication and compression are common methods for reducing the footprint of data in a storage system. Data Reduction Ratio (DRR) is a measure of the effectiveness of data reduction. It is the ratio of the size of data ingested to size of data stored.

$$\text{DRR} = (\text{size of ingested data}) / (\text{size of stored data})$$

Here *size of stored data* is the *amount of usable storage capacity consumed* by the data.

When it comes to an all-flash storage system like Pure Storage's FlashArray, its data reduction is purpose-built for flash to achieve three main goals.

1. Reduce flash media capacity needed so that more can be stored in less physical space. That should reduce the cost of the raw capacity. That should mean less floor space, power and cooling are required. This brings down the cost of ownership, by reducing both acquisition and operating costs.
2. Reduce write amplification, an undesirable phenomenon in flash media that affects its usable life span. Data reduction mitigates this problem by reducing the number of write operations.
3. Mitigate the imbalance between read and write performances for underlying NAND cells in flash media. Unlike disk media, the write performance density (IOPS/GB) for flash is much lower than its read performance density when addressing a large pool of flash tucked behind a microcontroller (e.g. SSDs). Data reduction also mitigates this problem by reducing the number of write operations.

Although Moore's Law is bringing down the cost of raw flash, #1 above continues to be a major benefit as organizations will reduce their overall TCO. As more and more bits are stored per NAND cell, both endurance (#2) and write performance density (#3) will get worse. Hence, good data reduction continues to be a critical requirement to extract the most value from an all-flash array.

Pure Storage has set the gold standard in data reduction for all-flash arrays. Our competition has largely

responded, where even possible to implement, by forcing data reduction technologies into their architectures that were originally designed for disk or hybrid systems. These 'retrofitted' designs attempt to use data reduction primarily to solve for #1. But these last-gen data reduction technologies fall short. Pure Storage's true variable block data reduction with 512byte level scan often delivers 2x *better* data reduction when compared with the leading competitive products. True variable block size chunking means the engine catches and dedupes data that fixed block size implementations fail to identify. And, scanning every 512 bytes means the process is auto-aligned with the applications, no need to configure or tweak block sizes. When competing products fail to meet the high bar set by Pure Storage, these vendors often attempt to distract customers with one of the following tactics.

Tactic A for data reduction: Make up for poor DRR with 'free' raw flash capacity

The vendor offers additional *raw flash capacity* to make up for the shortcomings in data reduction. They are hoping to convince the customers that DRR is just about improving the upfront cost of an AFA and these vendors are willing to eat the additional cost of giving away raw flash capacity to deliver the required *effective capacity (the data storage capacity a customer needs)*. Things to watch when reviewing their proposals are:

- Does the vendor downplay the value of DRR? Are they limiting the benefit of data reduction to upfront price savings and largely ignoring TCO?
- How does the vendor's proposal accommodate your data growth after the initial 'free' raw capacity offer? Do the initial savings cover the increased operating costs and maintenance for the 'free' raw capacity over time?
- Does the vendor limit its coverage for flash media (SSDs or Flash Modules) even when active maintenance and support is in place? Some architectures with no data reduction (or weaker data reduction) force the vendor to limit the coverage of flash media to fewer years or a fixed number of writes.

This tactic hides the fact that the flash media in a retrofitted AFA with poor DRR may have a shorter life span and reduced write performance. As storage needs grow, customers often end up spending more on flash media, both in terms of capacity purchases and operating expenses, after the initial 'free' offer.

Tactic B: Loosely 'redefine' data reduction to artificially inflate DRR

The vendor stretches its definition of DRR to cover capabilities outside of data reduction - ones that do not address #1, #2 and #3 above. The hope is that customers will simply look at their artificially inflated DRR numbers and assume that their retrofitted AFAs will provide DRR, performance and reliability equivalent to Pure Storage's FlashArray. Here are a few tricks seen in the wild:

- **Thin provisioning** - Many vendors include perceived storage savings from thin provisioning in DRR. They can make up any DRR for marketing purposes by bloating the DRR using the provisioned capacity to used capacity ratio. For example, if a 1000GB volume is thin provisioned and the current used capacity is 100GB, the DRR is inflated by 10 (i.e. 1000GB/100GB). Some vendors don't clearly disclose that they are baking savings from thin provisioning into their published DRR.

- **Snapshots** - Some vendors include perceived storage savings from snapshots in DRR. The logic here is that the source and snapshot volumes are effectively sharing the same blocks, are reducing capacity usage, and hence should be counted as 'data reduction'. This problem is similar to what you see in the thin provisioning benefit. You can artificially inflate DRR. N number of snapshots can inflate DRR by a N+1.
- **RAID erasure coding** - Some vendors, in particular hyper converged (HCI) system vendors, have recently started claiming storage savings due to erasure coding in DRR. Note that HCI systems have been keeping multiple copies of data across nodes for protection, thereby causing storage waste. Recently, HCI systems have started offering to replace default mirroring (optimized for performance) with RAID 5 erasure coding (optimized for capacity). RAID 5 is NOT a data reduction technique; it is a way to mitigate storage waste. However, HCI vendors are adding this 'storage waste avoidance' to inflate their published DRR.

Key takeaways

Don't be duped by inflated data reduction ratios. Don't blindly believe in programs or guarantees offered based on published DRR. Don't let a vendor hide the details at your expense - ensure you can make an apples-to-apples comparison. Ask the following key questions and get the answers in writing.

- Please define your data reduction ratio (DDR). Is there anything other than pattern removal, data deduplication and compression included in calculating the DRR?
- Does the user interface display DRR or equivalent metric without taking into account of thin provisioning, snapshots and RAID optimizations? Can you offer your data guarantee that is based on this 'true DRR'?
- I would like to talk to your customers who have deployed workloads similar to those in my environment. Can they talk about the DRR achieved without taking into account of thin provisioning, snapshots and RAID optimizations?
- Are there any trade-offs of using data reduction in your architecture? What is the impact of data reduction on your system performance? What happens if I also need encryption? Is data reduction available system-wide or do I need to manage it on a per volume basis?
- How long do you cover flash media wear? Do I need to rebuy flash media when the current flash media has hit its endurance cycle?
- How do you plan to ensure write performance density for your system as SSD capacities keep increasing? Will you provide a non-disruptive upgrade path when I need it?