Data Deduplication Explained

Data deduplication is often called "intelligent compression" or "single-instance storage" and can reduce storage needs by eliminating redundant data. Only one unique instance of the data is actually retained on storage media, such as disk or tape. Redundant data is replaced with a pointer to the unique data copy. For example, a typical email system might contain 100 instances of the same one megabyte file attachment. If the email platform is backed up or archived, all 100 instances are saved, requiring 100 MB storage space. With data deduplication, only one instance of the attachment is actually stored; each subsequent instance is just referenced back to the one saved copy. In this example, a 100 MB storage demand could be reduced to only one MB.

Data deduplication offers other benefits. Lower storage space requirements will save money on disk expenditures. The more efficient use of disk space also allows for longer disk retention periods, which provides better recovery time objectives for a longer time and reduces the need for tape backups. Data deduplication also reduces the data that must be sent across a WAN for remote backups, replication, and disaster recovery.

Data deduplication can generally operate at the file, block, and even the bit level. File deduplication eliminates duplicate files (as in the example above), but this is not a very efficient means of deduplication. Block and bit deduplication looks within a file and saves unique iterations of each block or bit. Each chunk of data is processed using a hash algorithm such as MD5 or SHA-1. This process generates a unique number for each piece which is then stored in an index. If a file is updated, only the changed data is saved. That is, if only a few bytes of a document or presentation are changed, only the changed blocks or bytes are saved, the changes don't constitute an entirely new file. This behavior makes block and bit deduplication far more efficient. However, block and bit deduplication take more processing power and uses a much larger index to track the individual pieces.

Hash collisions are a potential problem with deduplication. When a piece of data receives a hash number, that number is then compared with the index of other existing hash numbers. If that hash number is already in the index, the piece of data is considered a duplicate and does not need to be stored again. Otherwise the new hash number is added to the index and the new data is stored. In rare cases, the hash algorithm may produce the same hash number for two different chunks of data. When a hash collision occurs, the system won't store the new data because it sees that its hash number already exists in the index.. This is called a false positive, and can result in data loss. Some vendors combine hash algorithms to reduce the possibility of a hash collision. Some vendors are also examining metadata to identify data and prevent collisions.

In actual practice, data deduplication is often used in conjunction with other forms of data reduction such as conventional compression and delta differencing. Taken together, these three techniques can be very effective at optimizing the use of storage space.