IBM Data Footprint Reduction Technology

What is it?

Data footprint reduction is the process of employing one or more techniques to store a given set of data in less storage space. By applying data footprint reduction methods, a business can reduce storage costs and improve the performance of the backup/restore process needed to protect vital data.

What's the Need?

The amount of business-as-usual data (accounting, customer information, personnel records, email, etc.) being stored has grown steadily at a rate of between 40 and 60 percent per year. In addition to this growth, many organizations are seeing new data growth as they leverage IT beyond business-as-usual applications. Scientific applications such as genetic research, advanced imaging, weather prediction, and countless others drive more data growth. But this new data growth is not only caused by scientific applications... it also comes as businesses pursue new business models. Some energy companies have moved from manually reading their customers' power meters once a month to automated meters that report usage every 15 minutes. This one change represents a 288,000 percent increase in data. Web 2.0 companies such as FaceBook and Twitter are also seeing huge amounts of data growth.

During this time of exploding data growth, the cost of storage per GB has steadily dropped every year, helping to keep the cost of growing storing capacity in check. However, starting in about 2002, the storage cost curve started to become less favorable as disk manufacturers began running into the physical limitations of current storage technology (e.g., disk platter bit density).

It is this accelerating data growth—driven by business-as-usual applications, new applications, and whole new business models—combined with less favorable storage cost trends that has driven the need for data footprint reduction technology and its goals—to help reduce IT costs and improve data backup/restore performance.

How Does it Work?

Two popular techniques for achieving data footprint reduction are **deduplication** and **compression**. Let's take a quick look at how both of these technologies work.

Deduplication

The basic idea behind deduplication is that, in many cases, multiple copies of a given data item are stored on disk when only one copy should be sufficient. Such duplicate data items might include a digital company logo used in countless memos, a backup of a data file that has only been slightly changed since yesterday's backup, or a PowerPoint presentation sent to the inbox of many individuals around the office. Deduplication is accomplished by applying a **deduplication algorithm**, a mathematical formula used to guide the deduplication process on a set of data. The deduplication algorithm identifies duplicate blocks of data within data files, stores only one copy of each data block, and then keeps track of all other references to each of those data blocks.

As you might guess, deduplication works best when the data being processed contains a lot of duplicate data. While most situations will achieve some level of data footprint reduction through deduplication, one place to find a great deal of duplicate data is in the daily backups of important business information (what insiders of the IT industry refer to as **production data**) needed to recover from unexpected data loss. In most cases, today's daily backup data will contain many blocks of data within data files which have not changed since yesterday's daily backup run. Here is where you can often get the biggest bang for the buck with deduplication. In fact, deduplication arose from a desire to reduce the backup data footprint such that backups could be saved to faster but more expensive disk storage rather than tape storage. Backing up to tapes is inexpensive and works fine. However, in the event that a business needs to restore data from backups, restoring from tape can take a very long time due to the serial nature of tape media and the need to handle many individual tape cartridges... problems that only get worse as the amount of data increases. Deduplication is used as a way to address this performance issue by making it financially feasible for a business to backup to more expensive disk storage, which makes for a much quicker recovery after unexpected data loss. In actual business environments, a typical data footprint reduction from 8:1 to 12:1 can be achieved through deduplication of backup data alone. This means, for example, that 12 GB of backup data could be stored in 1 GB of disk storage space after deduplication, which represents significant savings in storage costs.

Another example of data that has a high degree of duplication is that found in the **virtual server** environment, such as those running IBM PowerVM, VMware, Linux KVM, Xen, or Oracle VM to subdivide a computer such that it acts as if it were multiple individual computers. These servers keep boot images of the operating systems for each guest and they are often nearly identical to each other - so deduplication can provide a good 80 to 90 percent (about 10:1) data footprint reduction here.

Now let's look a bit closer to see how IBM's latest deduplication technology differs from that used in other solutions. Traditional deduplication solutions use a **hash algorithm,** which examines a fixed-size chunk of data and performs a mathematical calculation that generates a **hash code** (i.e., a specific number) reflecting the contents of that chunk. It then compares that hash code to other hash codes stored in the **hash code table** (a constantly updated set of hash codes usually stored on disk). If the hash codes do not

match, then the new data is unique (not a duplicate) so that chunk of data is written out to the storage. Mathematically, two different chunks of data can generate the same hash code (known as a **hash collision**) so if there is a match, then additional processing may be needed to determine if the new data is in fact a duplicate of existing data. When a duplicate is found, it is discarded, and a reference or pointer to the existing data is recorded in the deduplication index (a table stored on disk that contains pointers to all the data stored in the deduplication repository). One down side of the hash code approach is that accessing a large hash code table can result in performance issues and increased storage requirements. These problems get worse as the amount of data being deduplicated increases. To reduce the likelihood of a hash collision, manufacturers have lengthened their hash codes over time and resorted to performing additional checking, resulting in very large hash code tables, additional performance problems, and increased storage space requirements.

The **IBM HyperFactor** deduplication algorithm (used in the IBM ProtecTIER family of virtual tape libraries for example) does not use hash codes, but instead examines a data item and stores some basic characteristics in a relatively small table residing in memory for fast access. It then compares these basic characteristics with those of previously stored data items. If there is a match, the algorithm then does a detailed comparison of the two items to see if they are indeed 100 percent identical. If they are, the new item is discarded and the second reference to that item is recorded in the deduplication index. In they are not identical, the new data item is written to disk storage and its characteristics are added to the index. The HyperFactor approach avoids the performance problems and expanding storage requirements associated with large hash tables. Instead, HyperFactor data deduplication uses a 4 GB memory resident table to track data item characteristics for up to 1 petabyte (PB) of physical disk storage. HyperFactor data deduplication is the reason IBM ProtecTIER is able to restore data from disk storage faster that competing solutions. And after all, restoring data faster is the primary reason for deduplication and storing backup data on disk storage in the first place.

Deduplication can be accomplished in one of two ways: **post-process** or **inline**. Post-process deduplication is done after a full copy of the data has been written to disk storage as a batch operation. This means that the IT infrastructure must have enough storage to hold a full copy of all data while it awaits deduplication. With inline deduplication (used in the IBM ProtecTIER family), data is deduplicated on the fly in real time as it comes in from the backup servers on its way to disk storage. This reduces storage space requirements since there is no need to store a full copy of the original data as with the post-processing method. In addition, inline deduplication in the backup data setting greatly reduces the amount of data sent over the wire to a remote location as part of a disaster recovery plan.

More on the Web
• IBM ProtecTIER (Inline HyperFactor deduplication) • "The Data Squeeze" from IBM Systems Magazine

Compression

Another way to achieve data footprint reduction is through compression. The idea behind compression is to identify commonly occurring patterns in data, represent those patterns using some type of short hand method, and then store the short hand version rather than the original data with the goal of saving storage space. Data compression is accomplished by applying a **compression algorithm** (i.e., a process of converting original data strings into shorter ones without losing information) to a set of data and storing the resulting compressed data rather than the original data.

Depending on the characteristics of the data and the compression algorithm in use, you will get variable amounts of data compression. Typically, compression results for either production data or backup data range from 50 percent (i.e., a 2:1 **compression ratio**) to 80 percent (5:1). So if you have a 10 GB file to store and you achieve a 2:1 compression ratio, you can store that file in only 5 GB of physical storage space, representing significant savings. Since applying a compression algorithm takes processing power and time, some compression solutions can reduce overall system performance.

In the right environment, the benefits of using either deduplication or compression are substantial. But can you use both technologies at the same time to achieve even greater data footprint reduction? Many compression algorithms tend to undermine the effectiveness of subsequent deduplication effectiveness. However, the IBM Real-time Compression algorithm is an exception, as it was designed to work well with deduplication.

Let's take a quick look at what makes the **IBM Real-time Compression** algorithm different from other solutions. Traditional compression algorithms take a given data file, break it into small chunks, and run these chunks through the compression algorithm, resulting in a rigidly-structured, compressed file of variable size (depending on original file size and compressability). Whenever the original data file is subsequently changed and resaved, everything in the compressed file after that change has to be recreated from scratch. This can impact overall system performance and result in compression ratios that degrade over time (due to disk fragmentation, garbage collection, etc.). In addition, since deduplication no longer recognizes anything downstream of the change, traditional compression undermines the effectiveness of the deduplication process, which winds up rewriting data that hasn't changed.

With IBM's Real-time Compression algorithm, a stream of data is run through the algorithm until the algorithm is able to produce a chunk of fixed size, organized in a file of flexible structure. Whenever the data is updated, only the modified sections of the compressed file are changed and so the file size remains the same, maintaining compression levels and improving performance. In fact, Real-time Compression often improves overall I/O performance because the overall amount of data written to disk is reduced. Furthermore, since most of the compressed file's contents typically remain stable, deduplication continues to recognize that unchanged content as duplicates, thus improving the overall effectiveness of deduplication.

The bottom line is that combining deduplication and compression can maximize an organization's return on investment and dramatically improve data protection performance and capabilities.

More on the Web
• IBM Real-time Compression Appliance • Video: IBM Real-time Compression technology overview (1:37)

Data Footprint Reduction at UPMC

"With ProtecTIER and XIV, we've been able to not only transform the environment and make it more stable, we've been able to take some of the manual housekeeping out of the picture and grow this staff-neutral."
- Kevin Muha, enterprise architect and technology manager of storage and data protection services, UPMC

University of Pittsburg Medical Center (UPMC) is an integrated global health enterprise headquartered in Pittsburgh, PA, and one of the leading nonprofit healthcare systems in the United States. Faced with storage demands that had grown by 328 percent in just three years, UPMC needed a new storage solution that could save data center floor space and also allow for faster, more reliable backups and restores.

UPMC deployed two 100-terabyte IBM System Storage TS7650G ProtecTIER Deduplication Gateways and three IBM XIV storage units to handle the organization's rapidly growing backup and recovery needs. The overall solution is managed with IBM Tivoli Storage Manager and powered by an IBM Power 595 server running IBM AIX. The IBM solution enables a 24:1 deduplication ratio for Oracle, cuts backup times by 20 percent while drastically reducing the need for traditional tape, and reduces recovery times by more than 50 percent.

More on the Web
• Case Study: UPMC pairs IBM ProtecTIER and IBM XIV solutions