

# INTRODUCING THE HIFN *BITWACKR*<sup>™</sup> B1605R

DATA DEDUPLICATION, COMPRESSION, ENCRYPTION AND THIN PROVISIONING  
FOR LINUX AND WINDOWS SERVERS AND APPLIANCES

## Table of Contents

1	BitWackr Data Reduction.....	3
1.1	Data Compression .....	3
1.2	Data Deduplication.....	3
1.3	Evaluate Primary Storage Deduplication Effectiveness in Real Operating Environments and in the Long Term .....	4
1.4	The Role of Human Behavior in Data Reduction .....	5
1.5	The Sequence in which Data Reduction is Performed .....	6
1.6	The Combined Effect of Deduplication and Compression .....	6
2	BitWackr Data Reduction Effectiveness .....	6
2.1	Primary/Application Data Deduplication .....	6
2.2	Interpreting Primary Storage Data Reduction Effectiveness.....	8
3	Data Reduction Effectiveness in Microsoft Windows Applications.....	8
3.1	Interpreting BitWackr for Windows Data Reduction Reporting .....	9
4	Data Reduction Effectiveness in Linux Applications .....	10
4.1	BitWackr Read Performance .....	11
4.2	BitWackr Write Performance and the Effect of the DR1605.....	11
4.2.1	The Performance Impact of DR1605 Hardware Assist .....	12
5	BitWackr for Windows Performance .....	13
6	BitWackr Design Recommendations .....	14
6.1	BitWackr for Windows.....	14
6.2	BitWackr for Linux .....	14
7	Summary and Conclusion .....	15

## 1 BitWackr Data Reduction

The Hifn Technology BitWackr reduces the capacity required to store a given amount of data. It does this through a combination of data deduplication and data compression.

BitWackr data reduction, unlike many other products featuring data deduplication, is not designed to deduplicate backup data. Deduplicating backup data is application-specific and dependent upon the data formats produced by backup software products.

The BitWackr reduces the amount of unstructured (non-database applications) data retained by an enterprise in volumes that contain primary (first copy) application and nearline data for which performance is not the highest priority. Some examples of this would be NAS user home directories, document directories, low-to-medium-activity application storage volumes (like Microsoft Data Protection Manager and SharePoint data), VMware template storage and active archival storage.

### 1.1 Data Compression

Data compression is a technique that re-encodes data so that it takes up less storage space. Compression is performed by finding repeatable patterns of binary 0s and 1s within data blocks meaning that the more patterns that can be found, the more the data can be compressed.

Capacity optimizing primary storage highlights the effectiveness of data compression because there is less redundant data available to deduplicate than with backup data.

There are "lossy" and lossless" forms of data compression. Lossy compression works on the assumption that data doesn't have to be stored perfectly. Much information can be simply thrown away from images, video data, and audio data, and when uncompressed such data will still be of acceptable quality. Lossless data compression is used when the data has to be uncompressed exactly as it was before compression. If you compress a block and then decompress it, the block is not changed. The BitWackr employs lossless compression techniques to ensure that data integrity is maintained as data is compressed and decompressed.

The metric employed in data compression is the "compression ratio", or ratio of the size of the original uncompressed block to the compressed block. For example, suppose a block of data before compression occupies 64 kilobytes (KB) of space. Using data compression, that block may be reduced in size to, say, 32 KB, reducing by ½ the amount of capacity required to store the data. In this case, data compression reduces the size of the data file by a factor of two, resulting in a "compression ratio" of 2:1 or a "data reduction percentage" of 50%.

Some data can be highly compressed while other data will compress very slightly or even not at all. The amount of compression experienced depends on the type of data and the compression algorithm employed.

Be alert to the use of the term "compression" when referring to data deduplication products. Sometimes you will see the terms "compression" and "deduplication" used interchangeably in the trade press. In BitWackr, deduplication and compression are two separate and distinct data reduction technologies that work in concert to aggressively reduce the amount of data that needs to be stored.

### 1.2 Data Deduplication

Block-based data deduplication is a technique that eliminates redundant blocks of data. In a typical deduplication operation, blocks of data are "fingerprinted" using a hashing algorithm

that produces a unique identifier for data blocks. These unique fingerprints along with the blocks of data that produced them are indexed, compressed and retained. Duplicate copies of data that have previously been fingerprinted are deduplicated, leaving only a single instance of each unique data block along with its corresponding fingerprint.

Some data can be aggressively deduplicated while other data will show little to no effect from deduplication. The level of deduplication experienced depends on the type of data being acted upon and the behavior of those storing the data.

Unlike compression, deduplication is done across all blocks as they are stored, so deduplication effectiveness actually improves over time as more blocks are indexed.

### 1.3 Evaluate Primary Storage Deduplication Effectiveness in Real Operating Environments and in the Long Term

The reason backup deduplication achieves a high data deduplication rate is that evaluations are performed over time and as more and more backup data is written. Similarly, primary or application data reduction also becomes more effective as more and more data is processed by the BitWackr.

Figure 1 was produced from data collected by a BitWackr customer using the product to store NAS data in an engineering environment. Note that data reduction effectiveness improves as more users store more data on the BitWackr volume. Just as in backup, the true test of BitWackr data reduction does not come from a single trial run, but rather from real-world tests using real-world data and covering an extended period of time.

Data deduplication for primary/application data is a new storage application that demands new evaluation processes. As in backup deduplication, a single test can't tell the whole story.

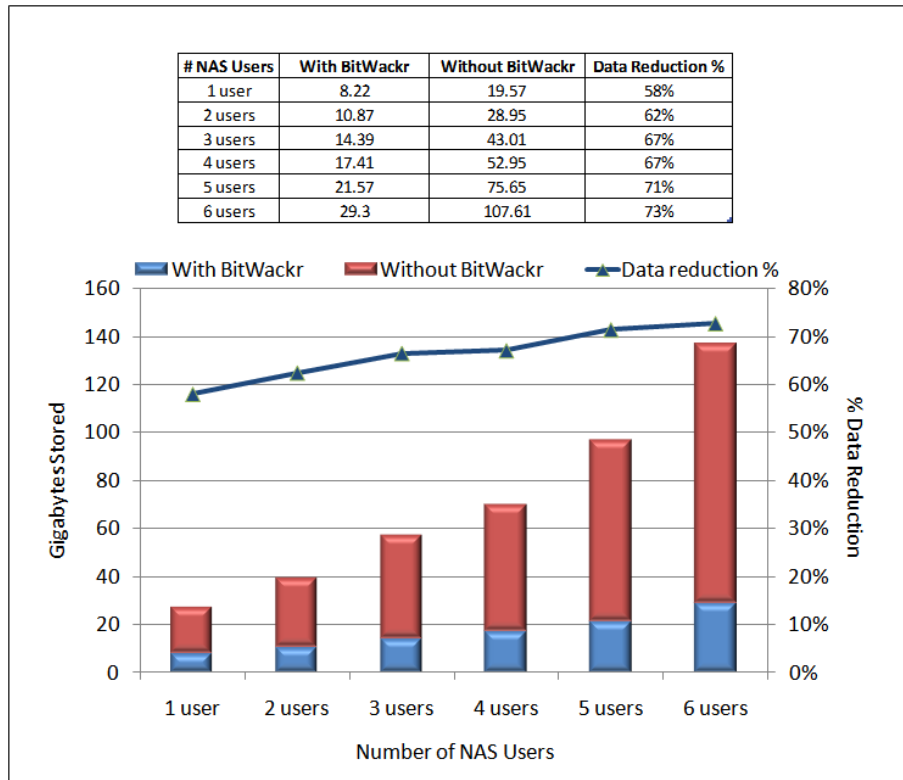


Figure 1. NAS Deduplication Effectiveness Increases as Additional Data is Ingested

## 1.4 The Role of Human Behavior in Data Reduction

Images, audio files, video files, scientific data acquisition, telemetry and geoseismic data types are typically random and unique and are most frequently compressed as they are created. Other unstructured data - like that produced by many business applications - is also typically compressible and can be compressed as it is written to disk. But whether data is produced by a person or a computational process, the behavior of the person responsible for creating the data plays a very minor role in the compressibility of that data.

On the other hand, data deduplication effectiveness can be heavily influenced by behavior whether intentional or unintentional. Several examples that may be illustrative follow:

1. An email with an attachment is received by 100 people in a company. 30 of the recipients want to retain the attachment for use at some time in the future so they store a copy of the attachment in their NAS home directories. If they are sharing the same NAS storage, there are now 30 copies of an identical set of blocks now stored on the device that are prime candidates for deduplication.
2. Mike manages a group of software engineering teams that are responsible for development, applications engineering, test/quality and technical support. Each team uses virtual server technology with similar operating systems for hundreds of virtual machines with just minor changes to each due to individual requirements, settings and preferences. These differences include hostnames, domain names, patch levels and network configurations. Given the amount of storage required for each virtual machine - yet the small differences among them - deduplication would greatly reduce the amount of storage required to support this virtual environment without impacting productivity.
3. Chris, a Microsoft Exchange administrator, has been tasked with migrating from MS Exchange 2007 to Exchange 2010 while integrating an additional 250 mailboxes because of a recent acquisition. Because he is being driven to minimize capital expenses, Nate is looking to reuse his data center's existing server and storage hardware. With the removal of Single Instance Storage (SIS) from MS Exchange 2010, Chris would benefit from deduplication technology to globally deduplicate data across old and new mailboxes.
4. Patricia works in the creative department of an advertising agency. She stores her work in folders on the department's Windows share. But Patricia isn't concerned with directory structures or conserving disk capacity. Her goal is to make sure that her illustrations are stored with their projects, so she makes copies of her graphics files to store with each of her projects. BitWackr deduplication can reduce the storage capacity required to hold those multiple copies.
5. Nate is the director of an IT organization that migrated from standard NAS data sharing to a Microsoft SharePoint environment in order to enhance collaboration on programs that span multiple departments. As time passed and programs expired, hundreds of files, documents, presentations, images and videos remain - many of which are still used repeatedly across multiple programs. Storing the SharePoint storage pools on a single deduplicating storage volume dramatically reduces SharePoint data capacity requirements resulting in reduced capital and operating expenses.
6. Chen runs an IC development group. He encourages collaboration among his team members. As a result of their collaboration, his team stores multiple copies of

documents, spreadsheets and CAD drawings that are either identical or very similar. Chen's NAS server is a perfect candidate for BitWackr capacity relief.

### **1.5 The Sequence in which Data Reduction is Performed**

The BitWackr combines compression and deduplication to reduce the capacity required to store data. Encryption is an additional administrator-selected option that can be invoked at the time a BitWackr volume is created.

In deduplication systems *other than* BitWackr, an incoming block is first hashed to extract its fingerprint. Next, in a second step, the block is compressed. Finally, in a third distinct step, the compressed block is encrypted. Note that compression always precedes encryption because the role of encryption is to introduce randomness into the data while compression operates best on data with the least randomness.

The Hifn DR1605 PCIe card – the hardware component of the BitWackr – performs SHA-1 hashing, eLZS compression and if selected, AES-256 CBC encryption *simultaneously* in a single operation. The SHA-1 hash is used to determine whether the block being processed is unique or is a duplicate. If the block is determined to be unique, the compressed (and optionally encrypted) block is stored in the BitWackr Data Store. If the block is a duplicate, appropriate counters are updated and the next block of data is processed.

Without the DR1605, compression, hash generation and encryption are performed serially in three time-consuming steps (which translates into latency). With the DR1605, these operations are performed simultaneously in a single pass, hence reducing latency.

By performing hashing and data transformation (compression and encryption) block operations simultaneously, the BitWackr reduces latency in the deduplication process. This is absolutely critical because latency is the enemy of deduplication system performance.

### **1.6 The Combined Effect of Deduplication and Compression**

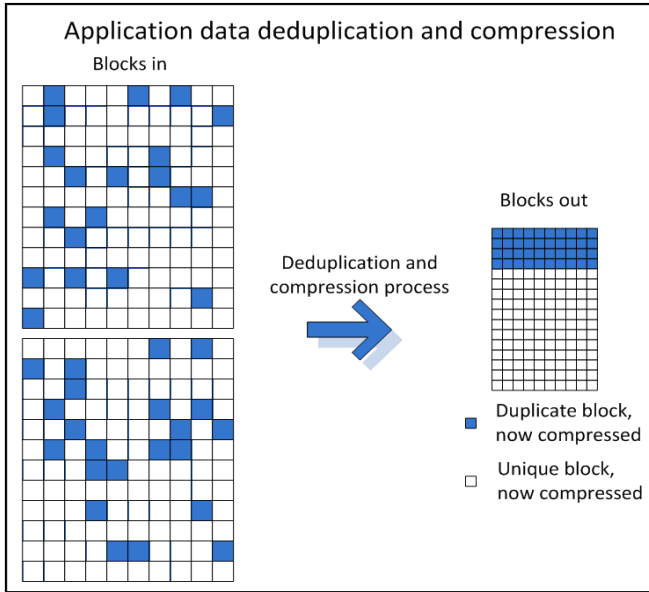
Data deduplication and compression work together to produce a combined data reduction effect. Depending upon the data, Hifn's deduplication algorithms can yield data reduction on the order of 10 to 80 percent for unstructured, application data. Compression works on the deduplicated data load as well as on blocks that do not deduplicate, shrinking the amount of capacity required to store data by as much as an additional 66 percent, so the combined data reduction and storage capacity savings over time can range up to as much as 90 percent (caution – your mileage may vary).

Our observations consistently show that on average about 2/3 of total BitWackr data reduction stems from data deduplication while the remaining 1/3 of the data reduction can be attributed to compression.

## **2 BitWackr Data Reduction Effectiveness**

### **2.1 Primary/Application Data Deduplication**

Primary data cannot be as aggressively reduced as backup data *because the same data is not repeatedly stored*, but as illustrated in the following sections, primary storage capital expenditure (CAPEX) and operating expense (OPEX) savings per gigabyte of capacity reclaimed are far more significant.



**Figure 2. Data Deduplication and Compression for Application Data**

We have found that BitWackr’s hardware-assisted eLZS compression plays an important role in primary storage data reduction. Primary storage data cannot be deduplicated as aggressively as backup data, making compression an important element of the BitWackr’s data reduction effectiveness.

The “before and after” results of data deduplication and compression for primary/application data are illustrated in Figure 2.

In application storage deduplication, new data is constantly being ingested at varying rates rather than in steady streams. This characteristic shifts the design point away from large segment matching to discrete block identification and classification. Application storage

deduplication requires extremely fast hash lookups (to identify duplicates) and high random write performance.

The need for random write performance may sound counter intuitive, but an example from the Hifn BitWackr may be illustrative. The BitWackr processes each block it sees as an independent event by simultaneously performing a SHA-1 hash operation, compressing and (optionally) encrypting the block. Blocks exit the BitWackr on their way to permanent media in sizes that are determined by the effectiveness of the compression operation. A block that was originally 64KB in size can remain a 64KB block or it can become a smaller, variable length block depending upon its compressibility.

These blocks are then stored in containers called “Extents” in the BitWackr Data Store that have been configured for variable length output block sizes, transforming a stream of sequential 64KB blocks into a collection of random, variable length I/Os.

The randomness in storing transformed data depends upon the compressed data block size and the availability and location of space within the Extents in which to store them.

In addition to randomizing the contents of the Data Store, the BitWackr’s deduplication metadata is also stored randomly.

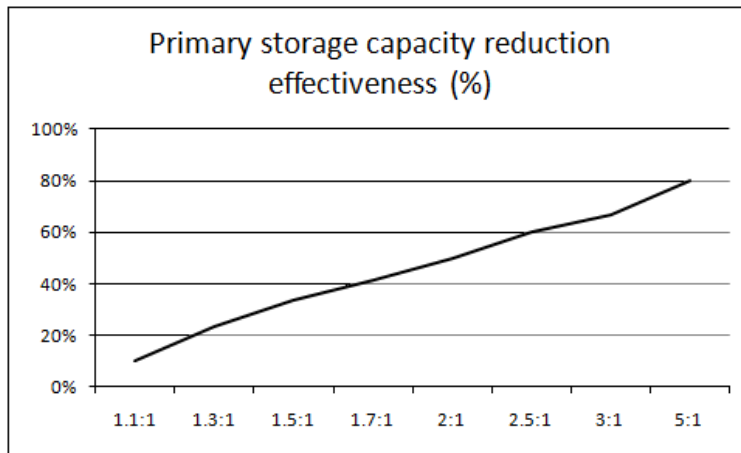
It is well known throughout the storage industry that random I/O throughput rates are lower than sequential I/O throughput rates, so sequential write performance can suffer from the effects of deduplication and compression. It is the job of the application storage deduplication architect to insulate the user from this effect to the greatest extent possible.

When using the Hifn BitWackr – particularly in a Windows environment - much of this performance insulation is provided by caching. Users operating with BitWackr-reduced data typically do not see much difference in performance from those not using the BitWackr.

Finally, backup deduplication products focus like a laser on writing data. Data is read only when a recovery operation is required following some sort of “disaster”.

To continuously optimize for writing data, some backup deduplication products perform “clean” operations (similar to a file system defrag) that force “older” data further away from the optimized data access path. Indeed, one industry leader suggests that the expected performance when accessing “older” data can degrade by as much as 60%. Other than backup, most applications read about 80% of the time and write for the remaining 20% making read performance extremely important.

## 2.2 Interpreting Primary Storage Data Reduction Effectiveness



**Figure 3. Primary Storage Capacity Reduction Effectiveness**

When examining the data reduction benefits of deduplicating primary or application data, capacity reduction effectiveness will be seen to vary by application and data type and will not typically reach the levels described by backup-specific products performing backup applications. But since application storage tends to be significantly more expensive to buy, maintain and operate than secondary storage on a cost/GB basis, primary storage

data deduplication can produce very significant overall savings.

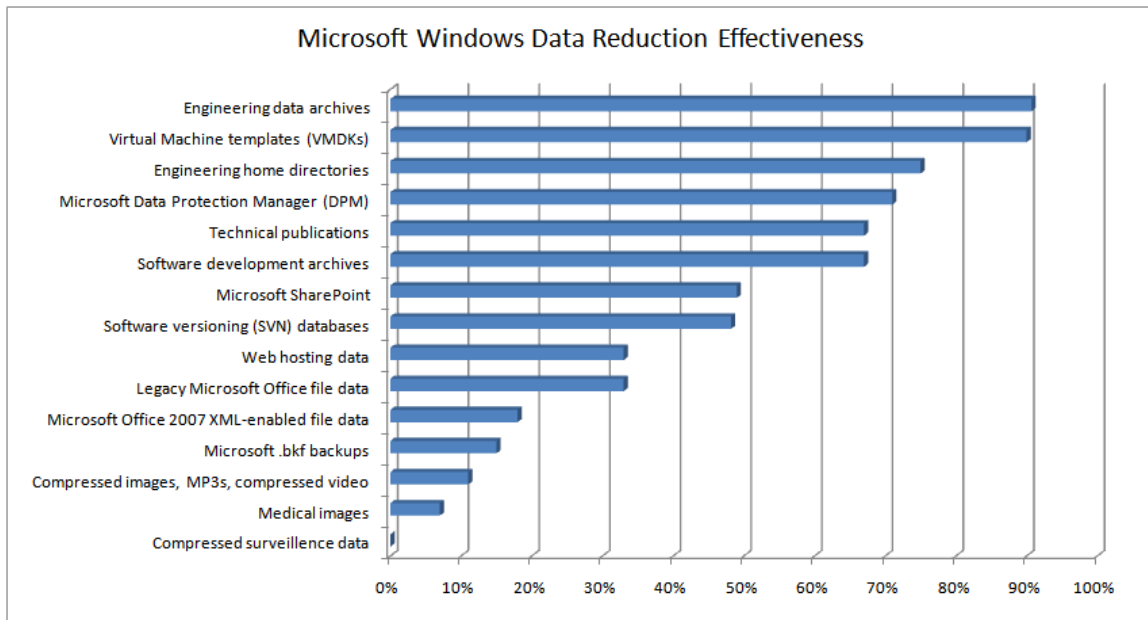
When dealing with data reduction for primary storage, the conversation moves from the “data reduction ratios” used to describe backup deduplication to “primary storage capacity reduction effectiveness” expressed in percentages as shown in Figure 3.

The BitWackr reports its capacity reduction effectiveness in three ways: as a percentage of capacity saved, as a data reduction ratio and as the number of megabytes of storage saved. These metrics are presented because relatively modest data reduction ratios (e.g., 1.9:1) tell us that almost half the capacity has been saved.

A data reduction ratio of just 1.5:1 - not considered very exciting in backup deduplication - represents a capacity savings of 33.3% - a considerable savings in both the CAPEX and OPEX of primary storage.

## 3 Data Reduction Effectiveness in Microsoft Windows Applications

The data reduction effectiveness test results summarized in Figure 4 were obtained from observations and measurements performed across a variety of BitWackr-equipped Windows servers using storage formatted with 64KB deduplication block sizes.



**Figure 4. Data Reduction Effectiveness in Microsoft Windows Environments**

The data reduction effectiveness results help to clarify the BitWackr’s market positioning. The types of data with the greatest propensity for data reduction by the BitWackr occur in NAS (NFS, CIFS) systems, document management systems, data archives and Virtual Machine template data stores.

The applications described above produce unstructured data that is retained by an enterprise in primary (first copy) and nearline storage volumes that are not performance driven.

Primary and nearline data that is not performance intensive is the BitWackr’s “sweet spot” and the area in which it delivers the greatest value.

### **3.1 Interpreting BitWackr for Windows Data Reduction Reporting**

BitWackr optimizes disk storage capacity utilization by reducing the amount of data that has to be stored. The key components of BitWackr capacity savings from data reduction are:

1. Capacity savings resulting from deduplication
2. Capacity savings resulting from compression
3. Total capacity savings resulting from the combined effects of data deduplication and compression provided by the BitWackr

Although both the Linux and Windows versions of the BitWackr report similar information about data reduction effectiveness, the data is presented in different forms.

Storage capacity used and available for both the virtual BitWackr volume and the underlying physical capacity is reported through the user interface.

Windows BitWackr system administrators view physical capacity utilization information for both the original and the BitWackr-reduced volumes as well as data reduction effectiveness through the BitWackr GUI’s “Capacity” tab as illustrated in Figure 5.

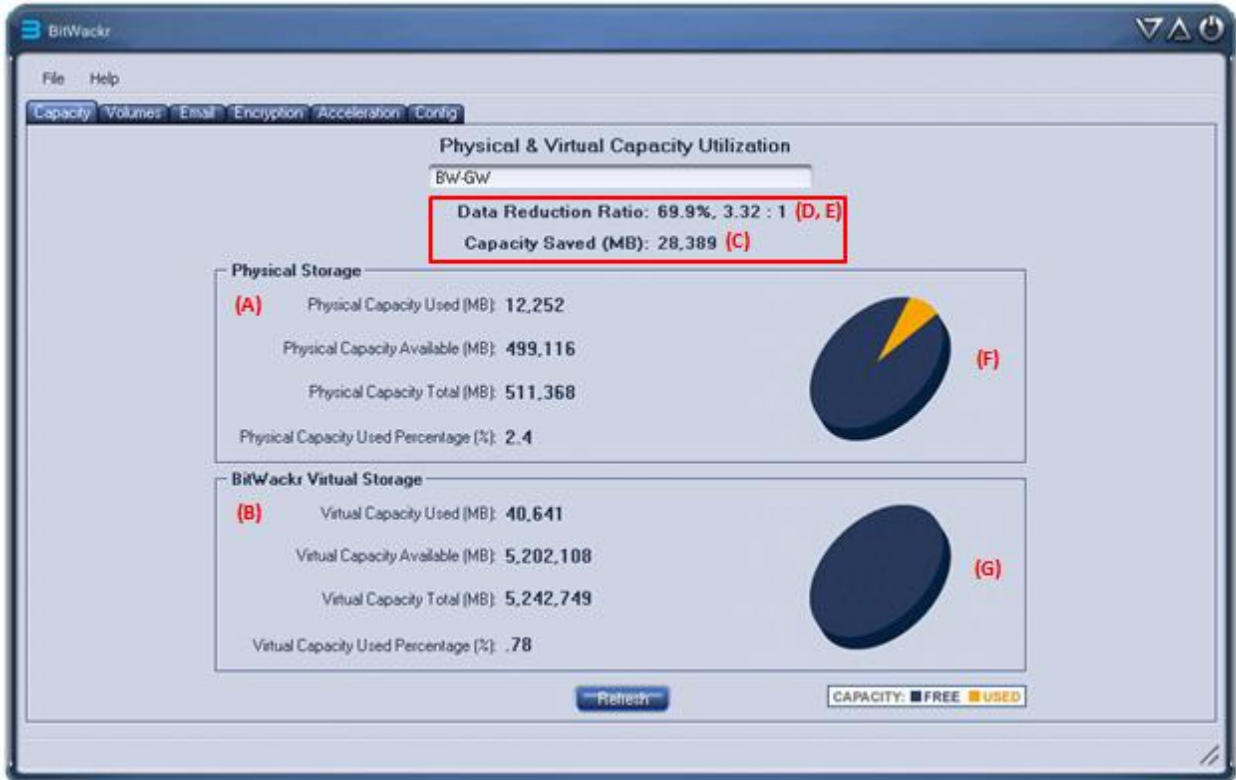


Figure 5. BitWackr for Windows Capacity Monitoring/Management Screenshot

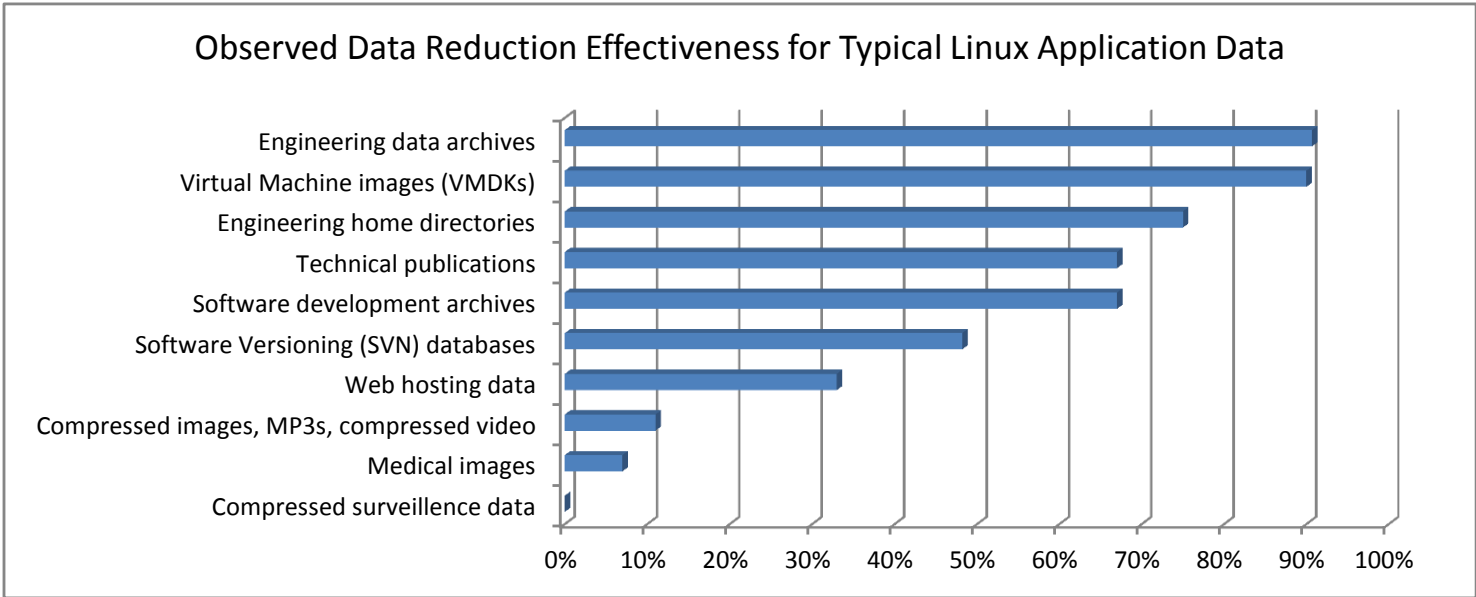
In Figure 5, a 5TB (virtual) BitWackr volume “BW-GW” currently consumes 12,252MB (11.96GB) of physical capacity (A). Without BitWackr data reduction, that data would require 40,641MB (39.69GB) of physical capacity (B). The physical capacity saved is 28,389MB (27.72GB) (C) which represents 69.9% (D) of the 39.69GB that would have otherwise been required.

Expressed as a data reduction ratio, this data is being reduced by 3.32:1 (E).

The pie charts on the right hand side of the GUI graphically represent the storage capacity savings and utilization (F, G).

#### 4 Data Reduction Effectiveness in Linux Applications

The data reduction effectiveness test results summarized in Figure 6 were obtained from observations and measurements performed across a variety of BitWackr-equipped Linux servers using storage formatted with deduplication block sizes ranging from 4KB to 64KB.



**Figure 6. Data Reduction Effectiveness in Linux Environments**

As in the case of the BitWackr for Windows results, the data reduction effectiveness results of BitWackr for Linux clarify the product’s market positioning. The types of data with the greatest propensity for data reduction by the BitWackr for Linux occur in NAS (NFS, Samba/SMB) home directories, all types of data archives and Virtual Machine template data stores.

The applications described above produce unstructured data that is retained by an enterprise in primary (first copy) and nearline storage volumes that are not performance driven. This is the BitWackr “sweet spot” and the place in which it offers the greatest value.

**4.1 BitWackr Read Performance**

Unlike backup deduplication systems, BitWackr has been optimized for read performance as described in Section 2.1. Applications other than backup on average read about 80% of the time and write for the remaining 20% making read performance extremely important.

Because of this optimization, data processed by the BitWackr can be retrieved from disk at rates similar to those observed in non-BitWackr-equipped systems.

**4.2 BitWackr Write Performance and the Effect of the DR1605**

BitWackr write performance (write throughput) is affected by a number of factors. The four points mentioned below have been observed as exerting the most influence over BitWackr write performance.

1. Metadata latency (most affected by the characteristics of the device used to store the Hash Table)
2. The presence of the DR1605 card (when writing compressed (and encrypted) data)
3. The level of deduplication being achieved
4. Single threaded vs. multi-threaded I/O

Additional BitWackr performance data is in the Hifn Technology document titled: “BitWackr Performance and Data Reduction Effectiveness” available under NDA.

#### 4.2.1 The Performance Impact of DR1605 Hardware Assist

To demonstrate the performance benefit of using the Hifn Express DR1605 PCIe card in deduplication, compression (and encryption), tests were conducted in which hash creation, compression and encryption were performed in software in a Linux test system constructed using an Intel Xeon 5500 Nehalem processor.

The server configuration used to perform these tests is illustrated in Figure 7.

BitWackr Performance Test Configuration	
Hash and LBA table storage	X Intel® X25-M SSDSA2MH080G1 2.5" 80GB SATA II MLC
Data Store	6x SAS HDD (15k RPM) – RAID 5
Hardware Acceleration	Hifn Technology Express DR 1605
Processor	Intel Xeon® 5500 (Nehalem @ 2.7 GHz)
Memory	3GB ECC RAM
Software (OS)	CentOS 5.3 (64 bit). Kernel 2.6.18 with Hifn BitWackr B1605RL for Linux software

Figure 7. BitWackr Performance Test Server Configuration

The results of this test are illustrated in Figure 8. While read performance is relatively unaffected by the Express DR1605’s acceleration (because decompression is computationally less intensive than compression), hardware-assisted write performance more than doubles over that possible when using software and a Nehalem general purpose processor for hash creation and compression.

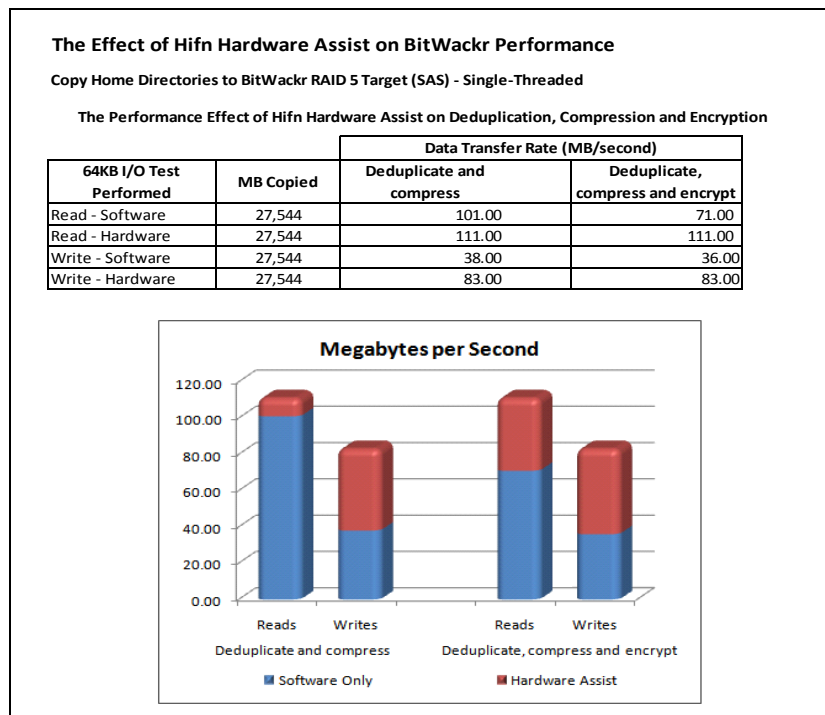


Figure 8. BitWackr Performance with and Without DR1605 Hardware Assist

The performance benefits of the BitWackr’s use of parallel processing in specialized hardware for data reduction and security becomes even more obvious when encryption is enabled. Figure 8 also illustrates that enabling encryption assisted by the Express DR1605 in BitWackr does not reduce either read or write performance. Indeed, Figure 8 shows that encrypted reads and writes achieve the same performance as unencrypted reads and writes, but only if the Express DR1605 is present.

## 5 BitWackr for Windows Performance

The deduplication block (cluster) sizes used by the BitWackr for Windows are typically tied to the file system managing the blocks (typically NTFS).

Windows NTFS is capable of producing blocks up to 64KB in size while the BitWackr is also capable of producing and managing deduplication blocks up to 64KB in size. For Windows implementations, we have observed that setting the deduplication block size in the BitWackr equal to the Windows NTFS cluster size at 64K provides an excellent balance between performance and data reduction effectiveness for typical Windows applications.

When examining the relationship between deduplication block size and BitWackr performance, the benchmark results presented in Figure 9 are helpful. Note that in this example (from a high-performance server and I/O subsystem) that both performance for both read and write operations peaks when using a 64K block size.

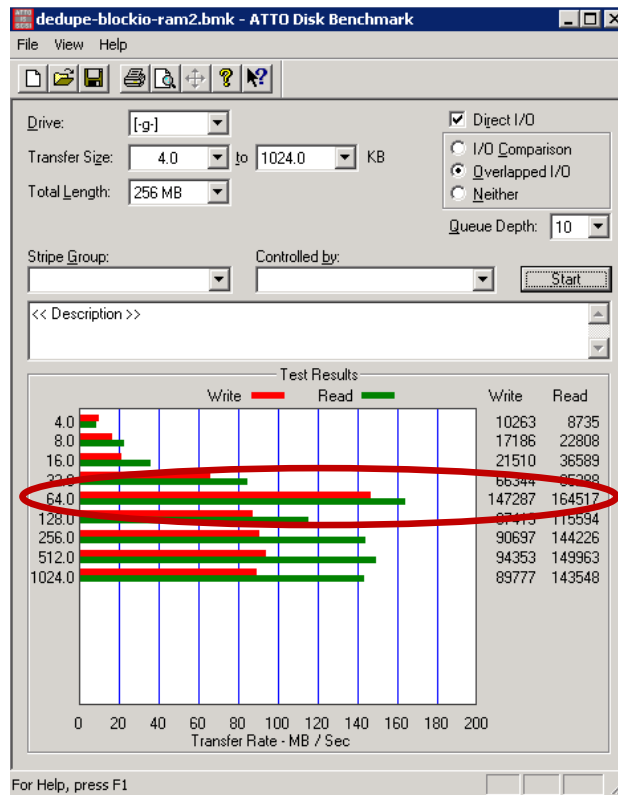


Figure 9. The Relationship between Deduplication Block Size and BitWackr Performance

## 6 BitWackr Design Recommendations

### 6.1 BitWackr for Windows

In this section we offer some design recommendations to OEMs, SIs and VARs building products incorporating the BitWackr for Windows.

1. We recommend the use low latency solid state disk (SSD) to hold the hash table. Since the hash table is most frequently accessed for reading (hash lookups), concerns over SSD write endurance with multi-level cell (MLC) SSD is of lesser concern, but single level cell (SLC) SSDs offer even better hash table performance plus higher reliability.
2. We have observed an excellent balance of performance and data reduction when using NTFS with 64KB clusters coupled with a BitWackr 64KB deduplication block size assignment.
3. Ideal application targets for BitWackr-equipped Windows servers and appliances include:
  - a) NAS (CIFS/NFS) shares
  - b) Office automation files (Microsoft Office, Open Office)
  - c) Disk-based Active Archives
  - d) Microsoft SharePoint data
  - e) Data Protection Manager (DPM – Microsoft’s CDP) Storage Pools
  - f) Microsoft Hyper-V virtual machine .vmdk file template storage
  - g) VMware virtual machine template storage; BitWackr efficiently stores VMware templates by deduplicating across all templates, both Windows- and Linux-based.
  - h) Deduplicating gateways
  - i) VMware application storage data reduction, capacity optimization and enhanced storage efficiency through in-line deduplication and compression.

### 6.2 BitWackr for Linux

In this section we offer some design recommendations to OEMs building products incorporating the BitWackr for Linux.

1. We recommend the use low latency solid state disk (SSD) to hold the hash table. Since the hash table is most frequently accessed for reading (hash lookups), concerns over SSD write endurance with multi-level cell (MLC) SSD is of lesser concern, but single level cell (SLC) SSDs offer even better hash table performance plus higher reliability.
2. Data reduction effectiveness is optimized when using 4KB and 8KB deduplication block sizes while data transfer performance is optimized when using 64KB deduplication block sizes. Using 64KB deduplication blocks in Linux is a recommended practice.
3. Ideal application targets for BitWackr-equipped Linux servers and appliances include:
  - a) NAS servers
  - b) iSCSI and FC storage target appliances
  - c) Disk based Active Archives
  - d) Deduplicating gateways
  - e) VMware virtual machine template storage; BitWackr efficiently stores VMware templates by deduplicating across all templates, both Windows- and Linux-based.
  - f) VMware application storage data reduction, capacity optimization and enhanced storage efficiency through in-line deduplication and compression.

## 7 Summary and Conclusion

Hifn's BitWackr is software combined with specialized hardware that enables OEMs, SIs and VARs to quickly and easily incorporate data deduplication, data compression, thin provisioning, enhanced security and physical storage capacity monitoring for capacity optimized volumes into Windows-based products and OEMs building Linux-based products.

The BitWackr is a developmental shortcut to the implementation of application data reduction products. The Hifn Technology customer implementing deduplication, compression, encryption and thin provisioning fully retains the ability to perform their own development and test efforts to deliver their own unique value-added functionality.

Hifn's BitWackr offers both the data reduction capability and performance needed to deliver high-value functionality to a wide variety of storage users.

The BitWackr is suitable for a broad range of both Windows and Linux applications and can be introduced into computing/storage environments with a minimum amount of effort.

Hifn Technology designed BitWackr with the objective of making the product extensible into more and more application areas and to meet even more demanding performance requirements. Much of the development team's efforts today are focused on further expanding into the Windows and VMware application markets. The team is currently looking at extending BitWackr to include Exchange email, VMware snapshots and other frequently-requested applications.

With the BitWackr, Hifn Technology is pioneering primary (application) and nearline storage data reduction. Using a combination of hardware-assisted compression and deduplication, the BitWackr is capable of helping storage administrators reduce their power consumption per TB of capacity while reducing rack space per TB, thus significantly lowering power, cooling and floor space requirements in the data center.

Because deduplication for primary and nearline storage is still an emerging technology, it is important to remember that some established test and evaluation methodologies may have diminished in relevance. Indeed, as with most new technologies, new processes and best practices will follow adoption.

BitWackr is on the leading edge of technology. And BitWackr today is the only technology available that enables the construction of the next generation of advanced data reduction products quickly, with minimal development and integration effort and at highly affordable price points.