



SiliconFS™

The BlueArc® Filesystem

Abstract

The BlueArc Filesystem, SiliconFS, is the engine which drives the entire architectural platform forward. The filesystem is the foundation of the BlueArc System Software which enables greater performance and scalability for the entire platform. The filesystem is what ultimately directs and manages that performance and scalability, harnessing the power of the BlueArc family of products to enterprise storage management features, providing real value for our customers.

Table of Contents

- A. Introduction1**
- B. Filesystem overview.....1**
- C. SiliconFS defined3**
 - 1. Object types4
 - 2. Checkpoints and NVRAM.....4
- D. Architectural advantages and benefits5**
 - 1. Fine-grained parallelism5
 - 2. Performance benefits6
 - 3. Resiliency8
 - 4. Dynamic storage expansion9
- E. Beyond performance and scalability10**
- F. Transparent Data Mobility 11**
 - 1. Intelligent Tiering.....11
 - 2. Data Migrator.....11
 - 3. Cross-Volume Links 12
 - 4. Dynamic Caching..... 13
 - 5. Data relocation.....14
- G. Data Protection 14**
 - 1. High-Availability design15
 - 2. Snapshots16
 - 3. JetClone.....17
 - 4. Replication features 18
 - 5. JetMirror 20
- H. Storage Virtualization21**
 - 1. Cluster Namespace and Mixed-Mode host support.....21
 - 2. EVS and Secure EVS support..... 22
 - 3. BlueArc Virtualization framework..... 23
- I. BlueArc Open Storage philosophy.....24**
- J. Future-proofing 25**
- K. Conclusions..... 25**

A. Introduction

BlueArc's history as a vendor of high-performance, highly-scalable network-attached storage (NAS) solutions stretches back more than ten years. Since its founding in 1998 BlueArc has consistently delivered on the promise of increasing the performance and scalability of its solutions with successive generations of products. Looking back over these many years of continuous development of the BlueArc System Software, the core architecture continues to evolve, offering increasing levels of performance and scalability at lower and lower total costs to the customer. And yet, the heart of the BlueArc architecture – the filesystem – also continues to grow and develop, adding features and functionality which enable greater utility for the customer as well. The BlueArc Filesystem, SiliconFS, is the engine which drives the entire architectural platform forward. The filesystem is the foundation which enables greater performance and scalability for the entire platform. The filesystem is what ultimately directs and manages that performance and scalability, harnessing the power of the BlueArc family of products to enterprise storage management features, providing real value for our customers.

SiliconFS is built around the Object Store, a collection of object structures referring to data on disks,¹ and a set of rules which govern the organizational layout and management of objects in the Object Store. The techniques behind creating, copying, moving, migrating, and deleting the objects in the Object Store make SiliconFS one of the most powerful, scalable, and extensible filesystems in use today. As the use of field-programmable gate arrays (FPGAs) to accelerate data operations is a key component in the differentiation of SiliconFS from every other filesystem in use today,² deciding which operations to accelerate in hardware plays an important role in the performance, scalability, and robustness of the filesystem's architecture.

SiliconFS also contains a number of enterprise storage features that distinguish it from competing products, with specific advantages in transparent Data Mobility, advanced Data Protection, and a rich Storage Virtualization engine to complement the high performance and scalability. While SiliconFS is itself proprietary, BlueArc maintains an entirely open philosophy when it comes to host operating system, network access protocol, and back-end storage manufacturer choices. BlueArc is not in the business of creating proprietary protocols or software to enable greater performance, scalability, or features, preferring instead an open storage path using agreed-upon market standards while fundamentally redefining the file server itself. The ability to have multiple storage tiers, centrally manage them, and simultaneously migrate data among them and to third-party storage platforms is another of the patented advantages of SiliconFS³ and proof of BlueArc's commitment to openness and standards.

SiliconFS has evolved over time to introduce new functionality and other improvements. This document describes to the latest version of SiliconFS.

B. Filesystem overview

SiliconFS is a highly differentiated technology that provides multiple benefits to its users. BlueArc products are best known for their ability to provide sustained, predictable, and consistent performance under various loads. SiliconFS is equally efficient with a variety of I/O sizes, loads, and access patterns. While superior single-server performance is important to many customers, SiliconFS also acts as an enabler for other important filesystem attributes:

- **Scalability without impacting performance:** SiliconFS can support millions of files in a single directory, while keeping directory search times to a minimum and sustaining overall system

1. At present both rotating disk and solid-state disk types are supported, as well as SSD/SDRAM hybrids; the dependence of the Object Store on rotating magnetic media is merely a functional definition.

2. Apparatus and Method for Hardware Implementation or Acceleration of Operating System Functions, United States patent 6,826,615 B2, granted 30 November 2004.

3. Network-Attached Storage System Device, and Method Supporting Multiple Storage Device Types, United States patent publication number W004/95287, application number PCT/US04/01352, filed 4 April 2003.

performance. Combined with Cluster Namespace™ SiliconFS can support many petabytes in a single unified namespace – presenting it all as a single filesystem accessible to many concurrent hosts, through a single mount point if desired.

- **Consolidation:** Extreme scalability enables consolidation, particularly of older hardware and “storage islands”. The ability to provide a unified, large-scale storage solution allows storage administrators to combine the functions of what were separately implemented file servers, reaping the cost-savings and ease-of-management benefits of a consolidated platform.
- **Meaningful virtualization:** virtualization is about making more efficient use of a single server. The more powerful the individual server is, the better suited it is to virtualize a larger number of less capable, under-utilized devices. BlueArc’s implementation of virtual servers allows groups to retain “ownership” of their virtual entity within a single physical server. And thin provisioning makes it possible for multiple virtual servers to share a single pool of storage devices.

SiliconFS offers benefits beyond sustained, predictable, consistent file server performance. Because of its unique architecture SiliconFS can adjust to the customer’s workflow and data sets. Not all data has the same “value” to the customer’s workflow. When data is first created it may be extremely valuable and must therefore reside on storage architected for very high performance. As the data ages application and eventually archival requirements tend to dominate, imposing further conditions on the storage where the data now resides. Yet, once an application knows where the data resides changes to the location are difficult. Applications and especially users do not like data migration. SiliconFS eliminates the difficulty of data relocation by providing a mechanism to migrate transparently across multiple tiers of storage, including data optimization devices (deduplication and compression, archival, etc.) Users do not necessarily need to be aware of the data actual location, and applications need not be rewritten either.

The following is a brief list of the key attributes of SiliconFS:

- **Widest applicability to changing workloads, data sets, and access patterns:** Fine-grained parallelism, off-loading of specific filesystem operations to FPGAs, and data pipelining all contribute to SiliconFS’ optimized handling of both throughput and metadata processing. Both attributes have been principal design criteria from the very beginning with SiliconFS.
- **Flexible performance scalability due to separation of function between servers and storage:** SiliconFS delivers great performance with relatively small storage systems. The filesystem also allows for performance to increase granularly as more disks are added. Typically this benefit will be felt immediately, even before “restriping” the data across both old and new spindles, as writes will automatically be spread immediately. As a result BlueArc customers may start small and scale performance by adding storage when needed. Additional file servers are not necessarily required for additional performance. As performance requirements grow even further, customers may also take advantage of clustering technology within SiliconFS to add more servers while maintaining a single namespace, providing easy management of very large pools of data. But again, SiliconFS offers true separation of function between storage and servers. Each may be scaled independently to meet a customer’s needs; there is no requirement to purchase one to get the other as with many competing NAS products.
- **Best-in-class namespace scalability:** Scaling beyond a single NAS server is essential for high performance storage solutions. Many parallel filesystem implementations rely on clustering multiple servers together for greater aggregate performance. The difference with SiliconFS is the scale: individual servers are much more powerful than traditional CPU-based architectures, meaning fewer servers are needed in a given cluster to achieve some specified level of performance. SiliconFS also makes it possible to create a single, unified namespace across the entire cluster of BlueArc file servers – making it appear as a single filesystem to all network hosts. This functionality is known as Cluster Namespace™, or CNS. CNS satisfies the most common scal-

ability requirements, allowing network hosts to access data on any BlueArc server in the cluster, regardless of physical location. SiliconFS takes advantage of BlueArc's unique architecture to move data seamlessly between multiple cluster nodes with minimal impact to performance.

- **Advanced multi-tier storage mechanisms:** Since data has an assigned value (by age, data type, owner, etc.) the ability to transparently relocate data to an applicable storage “tier” is a key feature of SiliconFS. Transparency requires that applications and users do not have to be pointed to new locations following data migration. SiliconFS provides policy-driven data migration mechanisms which allow data to be migrated transparently between many storage tiers. Individual storage tiers may also include 3rd-party, or foreign, filesystems accessible from the BlueArc servers via NFSv3 and HTTP. This ability to extend SiliconFS to external devices allows integration with many 3rd-party appliances for deduplication, compression, or archival for example. Such data migration mechanisms also allow for repurposing of existing storage devices as external storage tiers, lowering total costs and offering easier platform transitions.
- **Robust data protection:** SiliconFS provides various mechanisms for ensuring data protection. The storage used by the filesystem is protected by traditional hardware mechanisms, such as the use of redundant arrays of inexpensive disks (RAID) to provide fault-tolerance. SiliconFS also adds layers of functionality for further assurance of data preservation: enterprise features such as snapshots, replication, and high-availability cluster options are all part of the SiliconFS data resiliency framework.
- **Advanced storage virtualization framework:** A key advantage of NAS architectures over Storage Area Network (SAN) designs is the ability to more readily virtualize storage, simplifying data management and making storage provisioning much easier. SiliconFS provides an advanced virtualization framework that includes a global namespace (the CNS functionality), file server virtualization, storage pools, thin provisioning, and robust quota support.

C. SiliconFS defined

SiliconFS is implemented as an object-based design utilizing an object store, with root and leaf onode hierarchies in a tree structure, with a high degree of parallelization and manipulation of object pointers to accomplish data management duties. Most readers will be familiar with inodes, a data structure widely used in many UNIX or Linux-based filesystems. An inode stores information about a file, directory, or other filesystem object.⁴ The inode is thus not the data itself, but rather the metadata that describes the data. inodes store such metadata as user and group ownership, access mode (i.e., file permissions), file size, timestamps, file pointers (usually links to this inode from other parts of the filesystem) and file type, for example. When a traditional filesystem is created there is a finite upper limit on the total number of inodes – this limit defines the maximum number of files, directories, or other objects the filesystem can hold. This limit leads to what is called the finite inode problem, and is why most traditional filesystems cannot scale easily to multiple petabytes or billions of files.

In object-based filesystems objects are manipulated by the filesystem and correspond to blocks of raw data on the disks themselves. Information about these objects, or the object metadata, is called an onode, in much the same way inodes refer to file metadata in a traditional filesystem. In BlueArc's Object Store the underlying structure used to build up SiliconFS is an “object”, which is any organization of one or more of these raw blocks of data into a tree structure. The object is a container of storage that can be created, written to, read from, deleted, etc. Each element of the object is called an Onode, and while there are strong parallels to the normal use of the term onode in other object-based filesystems the concepts are not identical. In BlueArc's Object Store objects are manipulated by logic residing in FPGAs located on the hardware modules. SiliconFS achieves great acceleration of many filesystem operations through the use of FPGA hardware, and the design offers many performance, scalability, and robustness benefits to the end-user, such as:

4. inode data structure. <http://en.wikipedia.org/wiki/Inode>

- **Performance:** Writing Root Onodes to contiguous new space allows SiliconFS to take advantage of stripe set flushing, a technique designed to collate multiple writes into a single disk operation, thus obtaining maximum performance from the disks.
- **Relocation:** As Root Onodes are continuously written to new disk space, SiliconFS can move Root Onodes from their current positions if desired, allowing features such as volume shrinking and defragmentation.⁵

1. Object types

Different types of objects serve different purposes. Some objects, like the indirection object and free space bitmap, are used to contain critical metadata. Objects are used for other types of metadata as well, such as access control lists (ACLs). All critical metadata objects are automatically duplicated, when possible using different storage devices. SiliconFS contains a number of mechanisms that make it resilient to storage trauma, making it possible to recover from storage failures.

User data is contained in a file object. A directory name table object contains file and directory names in various formats (DOS short names, POSIX names, etc.), file handles, a CRC hash value, and the associated Object Identifier (OID) that points to the location of another object such as a subdirectory (another directory name table object type) or a file (the file object). Directory and file manipulation, snapshots, and other filesystem features benefit from this object implementation versus the more traditional file-level inode structure. A good example of this benefit is delivered via a unique object called the directory tree object.

For each directory name table object there is a directory tree object, although there can be many of the former in the latter. The directory tree object is a sorted binary search tree (BST) of Onodes containing numeric values (hashes). Converting the directory/file name to lower case and then applying a CRC algorithm against it derives these hashes. The benefit of this extra bit of metadata comes when it is time to find a directory or a file. When a host requests a particular file/directory by name, that value is again converted to lower case and the CRC algorithm is applied. FPGAs execute a binary search of numeric values (as opposed to having to do computationally expensive string comparisons of names) to locate the position within the directory name table object at which to begin the search for the required name. The result of this structure is a dramatic improvement in object lookup speed, providing a performance benefit to the end-user or application.

2. Checkpoints and NVRAM

Like many other advanced NAS platforms, SiliconFS uses a write-back cache to increase I/O performance while maintaining data integrity. All operations that modify the filesystem are also preserved in non-volatile memory, referred to as NVRAM. Writes and other modifying operations are not acknowledged to network hosts until the changes have been written to both the write-back cache and to NVRAM. Data is periodically flushed from NVRAM to disk as part of a checkpointing process. Checkpoints are taken periodically as a routine part of SiliconFS' normal operations, when the filesystem has consumed a certain amount of the NVRAM that has been allocated to it, and on certain other filesystem operations (e.g., when taking a snapshot of the filesystem).

At the end of each checkpoint, a consistent copy of the filesystem is located on disk. SiliconFS preserves the newest 128 checkpoints on disk. In the case of a system failure, such as loss of power, all transactions that have been acknowledged to network hosts are preserved either on disk or in NVRAM. Upon restart each filesystem is recovered before it is mounted and made available again to hosts. Filesystem recovery consists of selecting the most recent checkpoint and then replaying all filesystem operations that were logged since the last checkpoint, using the

⁵. Volume shrinking and defragmentation are not currently supported, but will be available soon in a future release.



information preserved in NVRAM. This restores the filesystem to the state it was in prior to the system failure, with no loss of data. In order to expedite recovery NVRAM replay is parallelized, and when possible multiple filesystem operations are replayed simultaneously.

In other failure scenarios, such as storage trauma or application-level failures, it may be beneficial to restore data from an older checkpoint or a snapshot. Filesystem rollback to any preserved checkpoint or snapshot is fast and easy, since no data relocation (copying) is actually required. Filesystem rollback is merely the rapid manipulation of object pointers in SiliconFS.

D. Architectural advantages and benefits

1. Fine-grained parallelism

The key to high performance in any filesystem is parallelism, and while SiliconFS can indeed be described as a parallel filesystem implementation there are striking differences when compared to other parallel filesystems on the market. The parallelism of SiliconFS is inherent in its design; it is much more than just a cluster of commodity hardware.

It is fine-grained parallelism which enables the extreme performance of SiliconFS. Historically, Multiple Instruction stream, Multiple Data stream (MIMD) architectures have been employed to attempt such parallelism.⁶ Traditional MIMD architectures, especially shared-memory implementations, require synchronization via a host operating system for memory coherence; this dependence often limits both overall performance and scalability. Modern MIMD architectures attempt parallelism by using a distributed memory design, using message passing or similar means to mediate synchronization issues. SiliconFS instead achieves fine-grained parallelism through its implementation in state machines, each of which control and enable specific functions. Two key features of this implementation which contribute greatly to the fine-grained parallelism are off-loading and pipelining.

Off-loading allows SiliconFS to independently process metadata and simultaneously move data to/from hosts and disks. Filesystem operations which do not require hardware acceleration through FPGAs are separated and sent to a metadata processor module, while operations in the data path are handled by a pipeline of FPGAs. Each filesystem path has dedicated memory, and in amounts specific to the operations required for that path. This off-loading is similar to traditional co-processor implementations (e.g., digital signal processors, systolic arrays, and certain graphics engines). Deciding exactly which operations are handled by which path is a crucial design characteristic for SiliconFS implementations.

In contrast, traditional shared memory architectures rely on CPUs for all filesystem operations and a single bus normally connects all CPUs to memory. In the SiliconFS design there is no single bus, and therefore no points of contention for memory access either. Even with distributed memory MIMD architectures still have bottlenecks (usually relating to message passing efficiencies at scale). The SiliconFS design avoids these issues as well: the filesystem paths are independent and do not require messages about data to be passed from one path to the other. While the metadata processor module is dedicated to data management the FPGA pipelines can focus on the business of moving data as quickly as possible.

Pipelining is achieved when multiple filesystem operations are simultaneously overlapped in their execution sequence. For a NAS system pipelining means multiple data requests (usually from some number of independent hosts concurrently) overlapping in the execution pipeline. SiliconFS achieves data pipelining by routing data operations to independent sets of FPGAs for accelerated processing. The operations are independent and have neither shared-memory nor message-passing dependencies.

6. MIMD architectures. <http://en.wikipedia.org/wiki/MIMD>

Manipulation of filesystem objects via the Object Store is central to SiliconFS' design, thus the benefits of extreme performance and massive scalability owe their existence to the fine-grained parallelism inherent in the architecture. Host access to the filesystem, however, is a different story. The Object Store is largely hidden from hosts, behind storage virtualization layers designed to make life easier for the storage administrator. Host machines have no concept of objects or the Object Store, and accessing SiliconFS via standard NFS or CIFS protocols they expect to work with string names and file handles.

For those hosts that require or prefer block level access, BlueArc also supports the iSCSI protocol, which requires presentation of raw blocks of storage to the hosts over an Ethernet connection. The host formats, and lays down its own (host operating-system-specific) filesystem structure upon, the blocks of storage presented to it. To support this within the Object Store structure SiliconFS creates a single large object of up to 256 terabytes in size (up to the maximum filesystem size) within the Object Store, which is presented as a sequence of logical blocks to the host. Since the iSCSI volume is just another object to the Object Store, features like Snapshots or dynamic growth of the entire object are possible, offering additional benefits over traditional iSCSI-based solutions.

BlueArc's Open Storage philosophy means that SiliconFS handles all conversion of objects to agreed-upon conventions for host presentation – namely the standards of the NFS and CIFS network filesystem protocols. This conversion is done transparently to ensure perfect compatibility so that hosts see only the standards-based representation of files. The BlueArc platform is not so much a traditional NFS or CIFS server as it is a parallelized filesystem engine presenting files to NFS or CIFS hosts in the manner in which they expect to see those files. This is another reason why the limitations of traditional NFS or CIFS servers really do not apply to BlueArc's architecture. Performance and scalability benefits are derived from the parallelism inherent in SiliconFS' architectural design while the “view” of what the hosts expect is a function of SiliconFS' rich virtualization layer.

2. Performance benefits

Beyond massive scalability in terms of overall data capacity, or in terms of billions of files, the SiliconFS design also enables two further performance benefits: high efficiency with a variety of I/O sizes and data access patterns, and near-linear scalability on I/O throughput with additional servers. Traditional filesystems are normally tuned for either small-block, random I/O workloads, or large-block, sequential workloads. Attempts to optimize filesystem performance for a wider range of application workloads normally involve the use of filesystem caches, read-ahead algorithms, adaptive schemes to avoid memory contention, etc.⁷ It is normally not possible to find a filesystem that works well for both small- and large-block data access patterns, or certainly not well for both at the same time.

Consistency of filesystem performance regardless of block size is an important feature and benefit of SiliconFS. Together with BlueArc's use of Intelligent Tiering for the creation and management of separate tiers of storage, storage administrators can design specific storage tiers for specific application workloads. Moreover, as the separate storage tiers can be united under a common namespace and could even be exported to hosts as a single mount point, SiliconFS can seamlessly merge both small- and large-block advantages into a single filesystem presentation to hosts. When contrasted with typical unoptimized filesystems, or even optimized filesystems with complicated caching and/or tuning workarounds, the advantages of SiliconFS' simpler design become clear.

7. UFS and NFS Cookbook, <http://nasconf.com/pres04/roch.pdf>, provides a good overview, albeit somewhat dated, of filesystem optimizations for Solaris. CITI Technical Report 06-04, <http://www.citi.umich.edu/techreports/reports/citi-tr-06-4.pdf>, describes typical filesystem inefficiencies of parallel filesystems in general along with a positioning of pNFS as a way to achieve optimization for both small- and large-block I/O patterns. Optimizing Input/Output Using Adaptive File System Policies by Madhyastha, et. al., <http://users.soe.ucsc.edu/~tara/pubs/goddard.pdf>, describes an adaptive process for optimizing filesystem performance based on continuous monitoring of application I/O patterns.

Because SiliconFS is not specifically designed for small-block, random workloads or large-block, sequential workloads, but happens to work well with either, the storage architect has more freedom in designing storage solutions. Confidence in SiliconFS to handle both small- and large-block workloads is backed by years of real-world data, and can be easily shown with straightforward tests, e.g., reports commissioned from The Tolly Group, a vendor-neutral benchmark validation organization. Using the well-known industry benchmark IOZone, The Tolly Group independently certified that SiliconFS delivers consistent results for both small and large block sizes. Copies of the report are freely available to anyone registering on The Tolly Group's website.⁸ More information on the open-source IOZone benchmark is available at the IOZone website.⁹

Storage I/O bandwidth, however, is only one measure of a file server's performance. Most storage architects concentrate on bandwidth as a measure of a system's overall performance because they assume the system will only deliver optimal bandwidth with one block size, or a limited range (usually very large block sizes only). Because SiliconFS can deliver excellent bandwidth with either small- or large-block workloads, and with just standard software clients (i.e., no proprietary parallel filesystem software), a better measure of system performance is I/O operations per second, or IOPS. Simplistically speaking, and absent other constraints, the delivered storage bandwidth is equal to IOPS multiplied by block size, so a truer measure of overall system performance is IOPS. Many storage vendors tend to shy away from pure IOPS benchmarks, preferring instead to state performance in terms of bandwidth and thus hide behind extremely large block sizes.

SPECsfs is the de facto vendor-neutral standard for all network file servers. Any storage vendor wishing to publically claim performance characterizations for their products must submit to the scrutiny of the SPECsfs benchmark; those vendors who do not submit data are quite often the ones hiding behind large-block data sets as a way to disguise poor IOPS performance. The implication of not submitting a SPECsfs benchmark is that the product in question is designed for large-block use only, or is not a performance-oriented network storage product at all.

BlueArc first submitted SPECsfs benchmark data in September 2004, with the release of the first-generation Titan server product. Since that time BlueArc has consistently been the highest rated vendor on the SPECsfs benchmark, with the fastest rating for any 1- or 2-node server configuration by any company.¹⁰ BlueArc's dominance of this benchmark has been used for many years to prove the superiority of SiliconFS' performance compared to traditional file server solutions, and is part of the reason BlueArc is able to deliver lower costs for a given set of performance requirements: faster performance per server directly translates to fewer servers needed, which directly translates to fewer devices to deploy, manage, license, upgrade, etc. and thus lower overall costs.

When clustering together any number of file servers, a certain amount of performance is lost to inefficiencies of the clustering process (usually as a result of increased communications between the various servers in the cluster, particularly for metadata operations). These inefficiencies are known as "clustering overhead" and can be measured as a deviation in measured performance from what might otherwise be expected to be a linear multiple of the number of servers in the cluster. That is, for n servers one might expect n times the performance of single server (call that P). If the measured performance is lower than the product of n and P , that difference is the clustering overhead.

8. <http://www.tolly.com/DocDetail.aspx?DocNumber=208351>

9. <http://www.iozone.org/>

10. <http://www.spec.org/sfs97r1/results/sfs97r1.html>

The level of performance lost to clustering overhead for SiliconFS is less than 0.6%. This compares to much larger overhead losses from other vendors when going from 1- to 2-node clusters, typically 8-10% even with the addition of a large number of additional disk spindles.¹¹ The benefit of near-linear scalability is clear, and reinforces the benefits of filesystem performance predictability over a wide range of data access patterns. The storage architect can design specific storage tiers and is confident the design will scale consistently with the addition of further servers. For a filesystem designed to scale to petabytes and many billions of files, near-linear scalability and performance predictability are important characteristics.

SiliconFS can distinguish between file system metadata and file user data. File system metadata is the “data about the data”, such as file attributes, permissions, access histories, and other descriptive or structural elements. The user data is what is typically thought of as the content of the file. Metadata is accessed far more frequently than the user data. Every time a file is opened, saved, closed, searched, backed up or replicated some portion of metadata is updated. Therefore when the metadata is separated from the user data and its access accelerated, overall system performance improves. Most competitors storage systems accomplish this by using expensive cache modules. This approach is simply throwing more hardware at the problem and does nothing to more efficiently handle the metadata.

SiliconFS offers superior value by including the multi tier file system feature for metadata management at no additional cost. While metadata is a small percentage of the total file system, the number of ‘metadata’ operations is many times the ‘regular’ data operations and contributes to a higher share of the I/O overhead. Since SiliconFS can automatically store metadata in high performance spindles or solid state drives while storing user data in low cost storage tiers it greatly reduces I/O operations overhead and enables cost efficiency with high performance. This simple capability results in optimization on several dimensions. Any combination of disks can be used SSD/SAS, SSD/SATA, SAS/NL-SAS, SAS/SATA and ultimately fewer spindles can be used to achieve the same levels of performance. When SSDs are used, only a small amount is necessary since space isn’t wasted storing and caching whole files.

3. Resiliency

Over the years, SiliconFS has added various filesystem mechanisms to tolerate different types of hardware storage failures and recover even in cases of catastrophic disk failure. One of the advantages of SiliconFS’ unique FPGA implementation is that data resiliency mechanisms that would have high performance impacts when implemented in software on traditional CPU-based solutions can be built into FPGA logic with negligible performance costs. Continuing engineering effort is being focused in this area so that higher levels of data protection prevent failures from occurring as much as possible, and recovery times are shortened in the event of failure in any case. Some of the data resiliency functionality currently provided by SiliconFS includes:

- **Protection of critical metadata:** SiliconFS protects all critical metadata via CRC checksums and end-to-end validators. Two copies of critical metadata objects are maintained, with each copy located on a different set of disks whenever possible. Failure recovery is implemented at a block level, making it possible to recover a failed filesystem even if both copies of the metadata structure are impacted: as long as one good copy is available for each individual block of data, the complete metadata object may be reconstructed from the constituent pieces.
- **Online consistency checking:** SiliconFS provides various mechanisms that check data consistency as background processes. These mechanisms are designed to detect various forms of failures, including unreported errors occurring at a disk level, or errors occurring internally to hardware RAID controllers. Although most failures of this type are extremely rare, the detection mechanisms built into SiliconFS ensure that the filesystem can react quickly to unexpected problems and either avoid or mitigate filesystem failures.

11. A comparison of SPECsfs97R3.0 submissions from various vendors confirms typical clustering losses for typical CPU-based architectures.



- **Versioning:** SiliconFS has the ability to “roll back” to previous complete checkpoints, usually the most recently completed checkpoint. By maintaining multiple checkpoints SiliconFS can also roll back further than the most recent checkpoint if desired – this ability is dubbed N-way Rollback. The effect is that the storage administrator can roll back entire filesystems to any arbitrary checkpoint that is complete, and very quickly too.

4. Dynamic storage expansion

An old storage aphorism says there are only two kinds of storage: new and full. Any filesystem must be able to deal with storage growth seamlessly or it will become very difficult to manage over time. The most difficult challenge when dealing with growing filesystems is maintaining consistent filesystem performance when new hardware is added (worst case) or increasing filesystem performance with the addition of new hardware (best case). Older filesystems which do not allow for dynamic expansion with new hardware require the storage administrator to either create separate filesystems on old and new hardware or copy the data off of the old hardware and back on to the combined hardware. The former path preserved existing filesystem performance but did nothing for increasing either performance or capacity of existing filesystems with the addition of new hardware. The latter could increase both performance and capacity of existing filesystems but only at the cost of a very laborious process involving large amounts of downtime. Dynamic expansion is the ability to automatically restripe data over both old and new hardware, without having to copy the data off and back on. The benefit of dynamic expansion to modern filesystems is to grow filesystems seamlessly while increasing both filesystem performance and total data capacity under management.

SiliconFS contains two separate but complementary features for dynamic storage expansion: Dynamic Write Balancing (DWB) and Dynamic Read Balancing (DRB). DWB distributes writes intelligently across old and new storage together. As new storage is added the DWB algorithm distributes new data across both old and new storage whenever a write operation occurs, taking care to balance both performance and data capacity (that portion of total usable capacity that is used for data). As the blocks of data are distributed across more spindles, performance increases. SiliconFS takes advantage of new spindles immediately but in most cases best performance is achieved when existing data is restriped across all spindles. For this reason DWB is not the complete dynamic expansion story, for it operates only with new data on write operations. For the complete story we need DRB as well.

A complementary feature to DWB, DRB utilizes DWB to complete SiliconFS’ dynamic expansion functionality. Whereas DWB can be thought of as an “always on” algorithm for write operations, DRB can be thought of as a “background process” which first reads and then rewrites data using DWB. When the DRB utility is started it begins rewriting files and stops once the data is balanced across all spindles. This process can take some time to complete if the amount of data to be restriped is considerable but eventually the DRB process will restripe and redistribute all data across all spindles in an automated fashion. Any hosts writing new data during the DRB process contribute to the balancing scheme.

A hidden benefit of dynamic storage expansion is SiliconFS’ ability to start small and deliver more and more performance as additional disk is added, and to do that cost-effectively. While the initial system performance may be short of the maximum possible, this ability to grow dynamically allows storage administrators to architect systems based on available budgets yet still “design in” total system performance as a function of predicted growth. This ability to scale granularly is hardly unique to SiliconFS, however delivering this benefit cost-effectively is another matter. The separation of function between the BlueArc server(s) running SiliconFS and the disk spindles underneath the filesystem is what enables this granular expansion capability on the most cost-effective basis possible. Contrast this benefit with what sounds like similar capabilities from other vendors and the difference is clear.

All clustered or parallel filesystem implementations contain the ability to scale performance granularly with additional disk hardware – this ability is one of the hallmarks of parallel filesystems in general. One may define “performance” simply or more narrowly, but most will agree that as parallel filesystems scale up performance increases. But as most filesystems do not divorce the servers from the disk attached to an individual server, total costs are greatly affected as the entire system scales up. With other filesystems the disk is tied to the server – it is not possible to add more disk without also adding more servers. Whereas SiliconFS gives storage administrators flexibility to scale performance (more servers) and data capacity (more disks) separately, other filesystems tie the two together and thereby increase total costs. Every additional server requires additional capital expense, additional licensing, additional support costs, additional rack space, increases power, cooling, and network port requirements, and generally adds to the complexity of management. If the same level of performance can be achieved with only the addition of more disk spindles and the unused potential of the existing servers, why add all that extra hardware?

E. Beyond performance and scalability

At its essential core SiliconFS represents the intellectual property of the company, not the hardware platform or the back-end disk architectures, although all parts are needed to form a complete solution. While disk technology (and specifically SAN architectures, as opposed to JBOD storage) certainly is very important for the high performance, scalability, and even robustness of SiliconFS, it is the filesystem which enables the greatest utility for our customers. Enterprise storage is about enterprise data management features, not merely going fast or scaling large. With more than a decade of continuous development, The BlueArc System Software provides a multitude of features designed to make BlueArc storage solutions easier to deploy, easier to grow, more tolerant of failures, and far, far easier to manage for a variety of enterprise user environments. Many of these features can be broadly classified into Transparent Data Mobility, Data Protection, and Storage Virtualization sections. Other filesystems have some of these features. A few may have a feature or two that the BlueArc System Software with SiliconFS does not yet have. But only SiliconFS can draw so heavily on features from all three filesystem pillars and combine them with industry-leading scalability and performance and use only open, agreed-upon industry protocols without proprietary software.

There are a host of features any network-attached filesystem must have to be considered useful for most enterprise environments. While unnecessary for advanced functionality, enterprise customers have come to expect and rely upon these basic features as part of the definition of a network-attached filesystem. Features such as host-side network connection protocols (most often NFS and CIFS), SNMP support, anti-virus support, basic backup services, even Snapshots and replication are today considered to be part and parcel of network-attached storage solutions. Much of the engineering development in BlueArc’s early years centered on the development of these basic features, and today BlueArc offers the full suite of basic features as an integrated part of the BlueArc System Software with SiliconFS.

Advanced features, on the other hand, distinguish basic network-attached storage solutions from true enterprise-class filesystems. The ability to manage data on many storage tiers simultaneously, to migrate it among or between tiers, and even to third-party storage solutions, means that the storage architect can design those tiers for specific application or business requirements, and can tailor specific storage technologies for each stage of the data lifecycle, and still retain the flexibility to use a very wide range of storage technologies throughout. The ability to centrally manage hundreds of disparate filesystems under a single, unified, global namespace, often with different (and concurrent) host connection protocols, means that the storage administrator can more easily manage a large heterogeneous user environment. The ability to automatically and seamlessly fail over server duties from one physical server to another means



that the storage architect can avoid unplanned downtime, increase storage utilization across servers, and even load balance to maintain optimum performance. Such advanced storage features are what distinguish the BlueArc System Software with SiliconFS from the majority of its competitors. Advanced storage features are not easy to do well; that is the reason many freely available or less-developed filesystems do not have them, or cannot make them work well at scale.

F. Transparent Data Mobility

Transparent Data Mobility (TDM) is a powerful concept in data management. The term refers to the movement of data along various points in the data lifecycle. All data has a point of origin (an instrument for example), and data may need to be moved to where it is initially used (e.g., heavy computational processing), and moved again to where it may more properly be classified for later re-use (home directories are typical), and finally managed data is deposited elsewhere for long-term archival. Different types of data may have different lifecycles or intrinsic value. It is entirely possible, even preferable, to design application-specific and/or user-specific tiers of storage for each stage of the data lifecycle. As multiple storage technologies are often the most appropriate match to each point in the lifecycle, the concept of tiered storage is central to an effective data management strategy. Certain tiers may be architected with difference performance characteristics in mind, or for better cost-effectiveness, or just so that the data they contain is bound to certain processes, applications, users, or groups. But simple storage tiering is not sufficient for an intelligent filesystem to deliver value to the storage administrator: for best value the ability to transparently move the data from tier to tier, keeping a single filesystem presentation to the hosts, users, and applications, is far more effective. BlueArc's term for data movement while maintaining a single filesystem view is called Transparent Data Mobility, and it has several components.

1. Intelligent Tiering

BlueArc has long championed the concept of tiered storage, and has for many years supported the use of multiple storage technologies underneath SiliconFS. In the early years of disk storage technology, the tiers were as simple as high-performance fibre-channel (FC) disks, and slower but more cost-effective ATA disks, and only one choice of vendor for each technology. Today those options have evolved into a number of choices of FC, SAS, SATA, SSD, and hybrid SDRAM/SSD products from a number of technology vendors, all sold and supported by BlueArc. While other storage vendors attempt to manipulate customers into just one or a few disk technologies (usually supplied only by them), BlueArc offers our customers a way to avoid vendor lock-in and expand their storage options. Today BlueArc sells and supports many choices of storage technology, allowing our customers to design a very effective and highly focused data management strategy using the most appropriate storage components for every point in the data lifecycle. The BlueArc term for this concept is Intelligent Tiering.

BlueArc's Intelligent Tiering allows customers to build scalable and flexible storage solutions that offer high levels of performance and cost-effectiveness with simplified and consolidated storage management. Using the various tiers of storage available, customers can keep data on-line longer without relying exclusively on tape technologies, minimizing the impact of backup, replication, or disaster-recovery requirements as the strategy requires. Intelligent Tiering gives data a longer disk lifecycle if desired, which can improve data access times for hosts and users.

2. Data Migrator

Merely offering choices of disk technology is not sufficient for an effective data strategy however. Once the storage architect defines two or more storage tiers, movement of data between the tiers becomes a critical design element. Specifically, policy-based movement of data between tiers is what makes the TDM strategy really effective. BlueArc's answer for this need is a product called Data Migrator. The simple description is that Data Migrator is the policy-based engine which allows storage administrators to implement their data movement policies. Data Migrator works

by allowing administrators to define policies, or even hierarchies of policies, which classify data and move that data from tier to tier based on criteria defined. Metadata attributes such as file type, file size, user or group ownership of file, last time of access, and dozens of other variables can be used to craft extremely effective data movement policies. Data movement may also be scheduled, running a policy check nightly, weekly, monthly, or whatever time period best suits the strategy. Different policies may be defined based on available free space, thus allowing for more aggressive migration policies when space is low. There is even a “what if” checkbox allowing storage administrators to craft a policy and analyze its impact on the various storage tiers, but without actually implementing the policy and initiating data movement.

Data Migrator solves one of the biggest challenges with out-of-band Information Lifecycle Management (ILM) solutions, a common problem with products from other storage vendors. When data is moved out-of-band, users must be notified of the new data location and applications have to be “reconnected” to the relocated files. Data Migrator is transparent to end-users and applications and does not require external ILM or data management devices. Because Data Migrator is an embedded feature of SiliconFS, all filesystem functions (e.g., Snapshots, replication, quotas, etc.) work seamlessly as if the data were still on the original storage tier and data integrity is maintained during the migration or recall. As far as end-users and applications are concerned the data has not moved at all. Users and applications see the data as if it still existed in the original location, while SiliconFS keeps track of where the data actually resides. For this reason BlueArc’s Data Migrator is often described by storage analysts as a “transparent, policy-based data migration engine” for implementing ILM policies. But here at BlueArc we know that Data Migrator is in fact the heart of the TDM concept.

3. Cross-Volume Links

Cross-Volume Links (CVL) and External Cross-Volume Links (XVL) are complementary technologies that extend the reach of Data Migrator. A cross-volume link is a zero-length file on a source filesystem (the primary filesystem) which “points” at a corresponding file on a target filesystem (the secondary filesystem). The pointer is stored in the Onode of the primary file. A flag in the Onode is used to indicate it is a cross-volume link rather than a regular file, and an extended Onode contains the information required to access the migrated file. All of the metadata required for directory level operations (including owner, access mode and ACLs) are maintained on the primary filesystem, so operations such as “ls -l” or “chmod” do not require access to the secondary filesystem. Similarly, the information needed for quota tracking is maintained on the primary filesystem, so quotas reported will include migrated files on the secondary filesystem as well.

The utility of the Cross-Volume Links to the TDM strategy becomes obvious once the storage architect migrates data to external storage devices. Cross-Volume Links are designed to operate either with internal BlueArc storage tiers or external, 3rd-party storage devices. It is the incorporation of external storage devices which greatly extends the reach of Data Migrator, and thus the entire BlueArc TDM strategy. Data can be migrated from tier to tier to tier, even to external tiers, and still be managed and presented to hosts and applications as a single cohesive whole. This is transparent, end-to-end data migration, a very powerful example of Transparent Data Mobility. While the use of external devices as remote target filesystems is currently limited to those devices which can be accessed via NFS or HTTP protocols, in theory future versions of the XVL technology could make use of additional protocols, greatly expanding the list of 3rd-party devices which could be incorporated into the BlueArc TDM strategy.

Repurposing of existing storage investments is another obvious benefit of TDM in general and XVL in particular. Every customer has some storage platform in use before they learn of BlueArc. Instead of throwing away that investment some customers may choose to take advantage of TDM features and repurpose that 3rd-party storage within the BlueArc namespace, per-



haps as an archival tier or even a crude replication target. While other vendors attempt to sweep the datacenter floor and encourage vendor lock-in, BlueArc would rather offer choices and ease platform transitions for customers.

Expansion of the BlueArc ecosystem of data management partners is another benefit of XVL. Current external XVL targets could be devices such as de-duplication and data-compression tiers, encrypted archive tiers (Vormetric), content-archive storage tiers (Hitachi's HCAP), or just about any 3rd-party device accessible from Titan through NFSv3. Data migration can also be controlled via an API that has been made available to selected BlueArc partners. The Hitachi Data Discovery Suite (HDDS) product, for example, uses this API for optimized search and indexing as well as to control data migration from SiliconFS to HCAP. As solution partners discover how to work with BlueArc's SiliconFS, more 3rd-party solutions will be incorporated into the data management framework, giving customers the capability of using both BlueArc and 3rd-party storage devices within a single, powerful, and transparent data migration strategy.

4. Dynamic Caching

Data Migrator may be the heart of TDM, but it is but one of several important features. It is the combination of such features that make the BlueArc's TDM design extremely robust and unmatched by any other ILM solution in the storage industry. Complementing Data Migrator are other features called Dynamic Caching and Data Relocation.

Dynamic Caching is a feature which reserves space on a storage tier for caching of "hot" files. The space reserved is actually an entire filesystem unto itself, and as such can be as large as any other filesystem in the BlueArc namespace. By definition, any file which is recently accessed may have a copy also located in the Dynamic Cache. If the cache is created in a high-performance tier of storage, this copy guarantees that any hot files are automatically on the highest performance disk tier (which may actually be an SSD or a hybrid SDRAM/SSD tier). Having the cache obviates the need for reverse data migration – why move the data back to the originating tier if a copy of it already exists on the highest performance tier?

Cluster Read Caching is Dynamic Caching applied to a cluster of BlueArc servers (i.e., many servers under a single namespace) or it may be applied to single BlueArc server. In the latter case the feature is called Local Read Caching. When used with a cluster of BlueArc servers, each server maintains its own Dynamic Cache, but is aware of the files accessed by all the other servers in the cluster. Copies of hot files from anywhere in the cluster therefore make their way to every cache on every BlueArc server, which can result in dramatic aggregate read performance improvements since every server can respond to any read request for a given set of hot files. In this way Dynamic Caching works with Data Migrator to provide policy-based data movement in both the forward and reverse senses simultaneously.

The read caching approach dynamically and transparently distributes and caches data to one or more designated data sets across individual BlueArc servers within a cluster. Policy-driven and fully automated, the Dynamic Caching transparently monitors file access patterns and caches only those files necessary to satisfy individual host and application requests received by SiliconFS. Customers with read-intensive workload profiles and a need to stage data in an optimized workflow process can leverage read caching as a way to scale performance when and how they need it. For many industries this capability translates to a common library of files, centrally accessed, which increases performance on-demand as additional hosts are added for applications which need to make use of the files in the library. Wherever storage systems are hitting hard limitations with performance or scalable and sustainable client/server access, dynamic read caching can help to achieve new levels of optimization and speed time to results.

5. Data relocation

Data relocation is the final feature of the BlueArc TDM design. Customers may need to relocate data for various reasons, e.g., optimizing workflow by moving certain data sets to a faster server or load balancing data across a number of servers. Three different data relocation mechanisms are provided:

- **EVS Migration:** (See next section of an explanation of the EVS feature.) EVS Migration makes it possible to relocate a virtual server within a cluster or to a server outside of the cluster that shares access to the same storage devices. EVS migration has minimal impact on network hosts and once it has completed those hosts may access the data using the same pathnames that were in use prior to the relocation. EVS migration is typically used for adjusting workflows or vacating a server for scheduled maintenance.
- **Filesystem relocation:** any filesystem accessed via Cluster Namespace can be relocated to another server within the cluster. Filesystem relocation has minimal impact on network hosts; once completed the data is accessed using the same pathnames that were in use prior to the relocation. Filesystem relocation is typically used to load balance within the unified namespace.
- **Data relocation:** data may be relocated from any given filesystem to another using a mechanism referred to as Transfer of Primary Access (TPA). TPA makes it possible to relocate individual directories as well as entire filesystems. TPA does however involve a small amount of downtime, and data is no longer accessible using the same pathnames that were in use prior to the relocation. TPA is generally used to better organize filesystems and/or directories within them.

G. Data Protection

Organizations with business continuity planning needs will recognize the importance of data protection features in their chosen storage platform. Much of the difference between enterprise storage platforms and solutions designed for the desktop, home, or small-business lies in these data protection features. All enterprise storage platforms have some measure of data protection beyond basic schemes like the use of RAID. Most enterprise systems are designed to continue operating even with major hardware failures; the system components are specifically designed with a high degree of fault-tolerance, and to contain zero single-points-of-failure, ensuring hardware redundancy at all levels. BlueArc System Software with SiliconFS goes beyond other enterprise storage platforms and also contains features designed to maximize system uptime, balance system load in real time, and even allow for maintenance windows without the need to take the system off-line. Beyond system robustness, SiliconFS also offer features for on-line data recovery, data replication, mirroring, backup, disaster recovery, and complete system monitoring capabilities.

BlueArc supports High Availability (HA) clustering of servers in a two-node Active/Active configuration or an N-way (more than two nodes) clustered configuration. Clustered servers provide NVRAM mirroring for enhanced data protection, automated filesystem failover, and higher levels of performance as additional servers are added to the BlueArc cluster.

SiliconFS provides additional mechanisms for data protection. Three of the more important mechanisms are snapshots, data replication, and data backup. Snapshots are generally described as point-in-time copies of the filesystem, and are a very convenient way to give end-users a way to “rollback” to a previous point in time to recover their own data. There are several data replication options within the BlueArc System Software; these may be described as either file-, object- or block-based, and synchronous or asynchronous. BlueArc provides a robust, flexible architecture for backup options as well. Snapshots, data replication, and data backup together provide the storage administrator with a range of choices for data protection. As with the choice of disk tiers, SiliconFS provides the customer with an architecture capable of selecting data protection options which are most suitable to the environment at hand.

1. High-Availability design

The foundation for BlueArc's High Availability (HA) design is the concept of Virtual Servers (EVS, for Enterprise Virtual Server). Virtual Servers are logical entities that reside on a physical server, in a manner analogous to operating system virtualization techniques such as VMWare, Microsoft's Hyper-V, or Citrix's XenServer. An EVS does not have physical interfaces per se, but instead has virtual interfaces that map to the physical interface(s) of the server. As a result of the separation between the physical and logical interfaces of the EVS and the actual server, an EVS can be migrated from one physical server to another transparently. In a failover situation for an HA cluster, this EVS migration is an automated process that can take place without system shutdown, and in most cases can occur quickly enough that hosts using stateless protocols (e.g., NFSv3) will not require unmounting and remounting of NFS exports.

Virtual Servers allow for centralized administration of the physical server and the various parameters which govern its operation, but give the storage administrator more flexibility to tailor to various applications, e.g., home directories, databases, backup duties, data migration, etc. Each EVS has storage dedicated to it, and access to the data is controlled by the Virtual Server Clients map and not by the physical server itself. Each EVS may have its own IP address, its own data management policies, and its own data exports/shares, and the assignment of these properties follow the EVS as it is migrated between physical servers.

Virtual Servers and EVS migration is central to the HA design of SiliconFS, but migration may also be useful in other scenarios. EVS migration may be used to load-balance operations between multiple physical servers in a BlueArc cluster, or Virtual Servers may be purposely migrated (i.e., manual failover) in order to clear a physical server of traffic prior to a maintenance window.

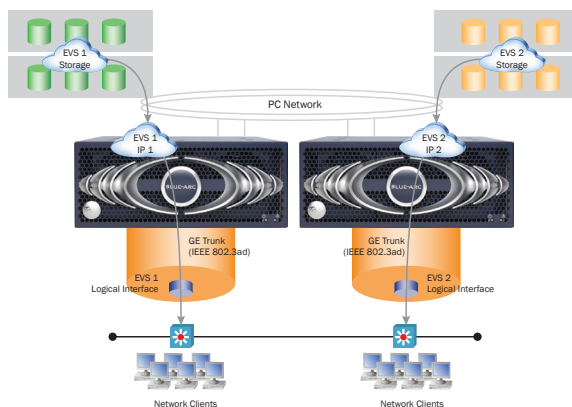


Figure 1: Virtual Servers configured in a BlueArc Cluster

If the physical interfaces of the server are trunked (e.g., multiple gigabit Ethernet interfaces together) all the defined Virtual Servers on the physical server use all of the trunked interfaces in a failover configuration. Individual physical interfaces may also be assigned to specific Virtual Servers, allowing a more granular level of throughput control for each EVS.

BlueArc servers communicate with each other over a dedicated, out-of-band, high-speed serial interface (HSSI). The HSSI is used by the servers to propagate the server's configuration (and EVS information) to each other, as well as for mirroring NVRAM data between the servers. Use of the HSSI link ensures that:

- The server configurations are always synchronized
- The surviving servers in an HA cluster can complete any outstanding data operation requests active in NVRAM. In N-way configurations, BlueArc servers mirror NVRAM to neighboring servers in a round-robin fashion. In the event of failure, the mirroring is re-established between the remaining servers.

Host writes are acknowledged only when the write has been committed to the server's battery-backed NVRAM. The servers also exchange a cluster heartbeat across the HSI, and a secondary heartbeat is also maintained via the sideband management network. This ensures that one server will not prematurely take over the functions of a failed server in the cluster. An independent management device, the Systems Management Unit (SMU), acts as a quorum device for even-numbered HA cluster configurations. (Odd-numbered HA clusters do not necessarily require a quorum device to ensure the cluster remains intact.) Use of quorum devices prevents split-brain conditions; fencing conditions are communicated cluster-wide through redundant data paths to ensure resource control and provide data integrity.¹²

SiliconFS buffers data in NVRAM until it is written to disk to protect it from failures as well as power loss. When servers are configured in a cluster, the servers mirror their NVRAM contents to other servers, thus ensuring data consistency in the event of failure of one of the servers. When the surviving server(s) in the cluster assume the functions of the failed server, they use the contents of the NVRAM mirror to complete any and all outstanding data transactions that were not yet committed to disk, providing seamless failover or service migration from one physical server to another.

2. Snapshots

Snapshots allow the storage administrator to capture a point-in-time of the filesystem – the point-in-time image is a read-only view of the filesystem. Using point-in-time images (snapshots), the storage administrator can:

- Allow end-users to retrieve files that have been deleted without administrator intervention
- Perform backups of the filesystem from a snapshot instead of using the live filesystem

Snapshots are rule-based, giving the flexibility to define them based on business policies. Rules-based snapshots provide entity management, a more useful configuration compared to simpler volume-based snapshot management. For example, hourly snapshot rules are managed as one entity, daily/weekly rules are managed as a separate entity, and monthly rules are managed as a third separate entity. There is also an implied hierarchy to snapshot rules: an hourly snapshot will not overwrite a daily/weekly snapshot, etc. An hourly snapshot will overwrite only another hourly rule, a daily/weekly snapshot will only overwrite a daily/weekly rule, etc.

The BlueArc snapshot mechanism works at the filesystem level using a pointer-based Onode approach that writes new data on new blocks in the filesystem, preserving the original data blocks. This method ensures no double-write penalty as seen with alternative copy-on-write snapshot methods. As with other snapshot implementations, BlueArc snapshots are block-based, meaning that only changed blocks of data are written to new locations on disk – the unchanged data is not moved and both the snapshot and the live filesystem can access the same unchanged data objects. This method vastly increases storage efficiency over file-based snapshot methods, which must copy the entire file to the snapshot location if any part of that file changes.

Different snapshot policies may be defined by the storage administrator for each filesystem. Should some data need snapshot protection and other data not, or if there is a large amount of data churn on a particular filesystem, snapshots can be turned on or off on a per-filesystem basis to manage the total disk capacity used by the snapshot feature.

Some key characteristics of SiliconFS snapshots are:

- Not all blocks are freed up upon snapshot deletion. Only those blocks which are exclusively linked to the deleted snapshot are deleted. Other blocks which may be linked to other snapshots are not deleted until all of the linked snapshots are deleted.



- With other storage solutions, snapshots can impose a significant system overhead, particularly when many filesystems are involved, or when a high degree of data churn is present. With SiliconFS snapshots are created and manipulated in hardware; there is no performance loss or additional system overhead on reads or writes.
- SiliconFS has aggressive object read-aheads to ensure high-performance read operations for all snapshot activities.
- Open files are snapped as point-in-time, i.e., last saved or last changed blocks, and may need to be coordinated with applications to ensure consistency.

The storage administrator may also configure the system so that snapshots are visible to end-users, or not visible, on a per-export basis. File and directory permissions associated with the filesystem are also preserved with the snapshot. This preservation allows end-users to access files and directories from snapshots and maintains the security associated with access rights to the volume. Because of this preservation the user cannot have permissions for the snapshot directory above what they have for the live filesystem – files or directories that the user cannot access to on the live file system are also blocked in the snapshot directory.

A snapshot of the volume may be taken in a number of ways:

- **Automatically:** via a prescribed rule
- **Manually:** using the servers' GUI or CLI management interfaces
- **Scripted:** using a script; the storage administrator may automate the snapshot process, in a manner similar to the rules-based method above
- **Event based:** using scripts and the BlueArc remote scripting tool, the storage administrator may automate a snapshot based on trigger events generated by the server.

The storage administrator is able to view what percentage of the volume is consumed by the live file system, and what percentage is used by snapshots. In order to ensure that there is always available space for snapshots on the volume, the storage administrator may set aside (reserve) disk space for snapshots, although such reservations are not required. The reserved space is dedicated for snapshots and the storage administrator can define the level as a hard or soft limit. Snapshots are stored within the related file system, so no space reservation is required unless the administrator wants to guarantee a proscribed amount of disk capacity.

3. JetClone

File clones provide the ability to instantly create space-efficient, writeable copies of single files. This is a key feature for virtualization, as it allows VMware administrators to quickly deploy new virtual machines, without consuming additional disk storage space.

When a file clone is created, a hidden file-level and read-only snapshot is created which preserves an image of the source file at that point in time. This snapshot allows the source file and clone to share common data blocks. If additional clones of the source file are created, new snapshots are created if necessary.

File clones can be created for both standard files and iSCSI volumes, with iSCSI support targeted largely towards supporting Microsoft application environments.

The key benefits of file clones include:

- Space Efficient - File clones are initially created through the use of pointers to blocks. No data is duplicated, and only new file writes are saved to the filesystem.
- Fast – File clones are created in seconds, regardless of the size of the file being cloned.
- Flexible – File clones are created per file, allowing for a VMware VM centric view of file cloning. No need to duplicate files that aren't needed.

Specific features of JetClone include:

- **Clone of a clone (aka Cascading Clones)** – This feature allows clones to be used as the source of a new file clone. Using a virtualization example, the VMware administrator has a ‘gold’ template of a Windows server. He clones that ‘gold’ template to create several application master templates such as an exchange server, SQL server, etc. Only the application specific data will be saved as new objects in the clone. Those application templates can then be cloned again and deployed as live virtual machines.
- **Deep Copy** – This is a standard copy, but it allows a clone that has diverged significantly from its snapshot to be broken out into an independent file as the user desires. Deep copy would also allow any outlying file clone snapshots to be deleted from the filesystem as a space saving measure.
- **No Read / Write Performance Impact** – Performance will be in line with the performance of standard file read / write. Auto-inquiry and auto-response for NFS and CIFS are fully supported.
- **No limits on the number of file clones** – There is no inherent limit to how many file clones can be created, as long as sufficient space exists on the filesystem to accommodate new data as it is created. This is a major consideration for virtual environments, as it will provide massive space savings for both virtual server and virtual desktop environments which share nearly identical OS images.

4. Replication features

Replication is the process of sharing data between redundant sources, as a method to ensure data consistency, and to improve reliability and accessibility of the entire system. Replication differs from backup in that replication aims to have the data in two or more places at once (theoretically, identical copies of the data in all locations at the same time), while backup aims to have two or more copies of the data at different points in time. Despite these differences there are many common design components in replication and backup, as both are designed with data movement in mind. SiliconFS provides several robust mechanisms for data movement, many of which are useful for replication scenarios.

Accelerated Data Copy (ADC) is a file-based, asynchronous method of data replication. ADC allows the storage administrator to define a policy-based data migration, or a mass data migration to occur either among or between BlueArc servers. Using Intelligent Tiering, ADC can move data among the various tiers of storage behind a single server in a non-intrusive fashion to the hosts connected to that server, thereby providing an easy, automated method for data migration between storage tiers behind a single server. ADC uses the Network Data Management Protocol (NDMP) to move data between servers. NDMP is an open protocol standard for enterprise-wide backup of heterogeneous network-attached storage. The Data Migrator feature of the BlueArc System Software and SiliconFS leverages ADC to manipulate data either intra- or inter-server as well.”

Incremental Data Replication (IDR) is an optional replication feature of SiliconFS. Replication occurs at the file level, and only files that have changed since the last scheduled replication are replicated. Multiple schedules may be defined on a per-EVS basis with support for pre- and post- scripting, enabling automated functions to occur prior to and immediately after the IDR schedule. A useful example is the automatic quiescence of a database prior to the IDR and then return of the database to an on-line state after the IDR process.

IDR also uses BlueArc’s ADC (NDMP basis) for data movement. IDR uses snapshots as a basis for replication, maintaining the last snapshot as the reference point for the next replication to occur to detect and track changes to files. Any delete, move and/or link operations occurring between the two snapshot references is replicated on the destination volume. Using snapshots as a means for replication also allows files that would normally be skipped because they are in use to be replicated, ensuring that all files in the volume/directory are replicated and protected. This offers a benefit over traditional replication schemes which may skip open files during the replication cycle.

Incremental Block Replication (IBR) allows storage administrators to set up a scheduled, incremental backup of volumes. Replication occurs at the block level, and only data blocks that have changed since the last scheduled replication are replicated. Block-level replication is extremely efficient, particularly if a volume has relatively small changes for large files, and may be more bandwidth-friendly compared to file-based replication schemes. IBR may be either synchronous or asynchronous depending on the configuration. As with IDR, multiple schedules may be defined on a per-server basis with support for pre- and post- scripting, enabling automated functions to occur before and after the IBR schedule.

Depending on the disk technology being used, BlueArc also supports specific block-based replication options available at the RAID controller level. Certain RAID controllers support controller-to-controller block mirroring (synchronous or asynchronous). Normally dark fiber connections between controller pairs are required for controller mirroring; depending on class of hardware and features selected, replication distances can range from 500m to 10km. Through the use of advanced fiber-optic wavelength-division multiplexing technologies, this range can be considerably extended to 100km or more.

As opposed to replication, backup services seek to achieve a copy of the data at a specific point in time, usually for off-line preservation. As with replication, SiliconFS leverages snapshots to allow the storage administrator to perform backups while continuing to serve data to hosts with the live filesystem.

SiliconFS supports NDMP versions 2, 3, and 4 for backup, and has an NDMP client built into the server as well. SiliconFS supports LAN-free backup of data using either FC or Ethernet networks and dedicated connections with dedicated bandwidth. This separation allows the live filesystem to continue serving data to hosts unimpeded by the backup operations. NDMP-compliant FC-attached tape libraries are recommended for most efficient use of available connections and bandwidth when designing backup solutions. As the host Ethernet network is not used in FC-attached backup scenarios, the storage administrator may decide to run backup operations at any time instead of waiting for periods of low bandwidth utilization on the Ethernet network (usually at night when users are less active), thereby greatly increasing the available backup window.

Beyond high-availability, snapshots, replication, and backup, BlueArc's call-home monitoring service provides a further level of data protection capability in the form of predictive failure analysis, as well as useful statistics for historical system usage and troubleshooting. The call-home service complements business continuity features such as high-availability, snapshots, replication, mirroring, and backup. The call-home service collects data from each BlueArc customer which allows such collection, and relies upon the Systems Management Unit (SMU) at each installation.

The SMU is a dedicated and integrated device which helps manage, configure, and monitor BlueArc systems. In its present form the SMU is a 1U, rack-mountable device integrated into the BlueArc solution; future versions will offer the ability to be integrated directly into the BlueArc servers or the SMU may be virtualized via VMWare or similar virtualization software and hosted on any customer server.

The SMU operates on the sideband management network of the BlueArc servers and does not impede the transfer of any data between the hosts and BlueArc servers. Rather, the SMU only deals with management functions for the system. Conceptually the SMU can be thought of as an integrated syslog server, collecting management data from all devices (BlueArc servers, FC fabric, and storage controllers) and presenting the storage administrator with a single plane of glass for unified systems management.

5. JetMirror

BlueArc JetMirror provides high-speed, object-based replication. JetMirror is a major new feature which greatly enhances BlueArc's data replication and disaster recovery capabilities.

The BlueArc SiliconFS filesystem is an object based filesystem, in which core filesystem structures and user data are stored as objects, rather than files or blocks. Object replication takes advantage of this structure to replicate these blocks natively, regardless of which file or directory that they may belong to, which negates the need to assemble all of the objects associated with a file before transfer, making the overall transfer more efficient.

In the case of incremental replication, object replication can efficiently compare the object changes between 2 snapshots, and quickly begin data movement. File replication must walk the filesystem and process the file metadata in order to determine which blocks to transfer. For dense filesystems, this can take hours using file replication, where block replication will take minutes.

Object replication is limited to running on the full filesystem, rather than directories or files, due to the nature of how the objects are moved. Trade-offs between object and file replication are discussed below.

The key benefits of object-based replication include:

- **Replication Performance** – Full replication performance will generally improve 2x – 3x depending on the structure of their filesystem. The greatest performance improvements will be seen in the incremental replication, especially for customers that have dense filesystems (millions of small files) or those that have a high rate of change in the filesystem. BlueArc testing has shown a 26x improvement in replication performance for just over a million files changed. Larger filesystems should expect even greater improvements in incremental performance.
- **Improved disaster recovery** – Covered in more detail below. Object replication enables the ability to quickly failover in the event of a disaster.
- **Flexible deployment options** – Object replication maintains the replication status on both the source and target filesystem. If the replication relationship is broken, such as during a system shut-down or move, when the relationship is re-established, incremental replication can continue, rather than requiring a full re-sync of the filesystem.

It is important to understand when to use object-based replication or file-based replication. Object-based replication is very focused on providing high speed disaster recovery, and opts for speed over the granularity that file-based replication provides. Both have value depending on the customer's needs.

- **Disaster Recovery (DR) or Backup** – Customers using replication for disaster recovery and business continuance will see major benefits from object replication and the disaster recovery features that it enables. Customers who are using replication as an additional layer data protection may want to use file replication due to its ability to recover a more granular subset of files.
- **Large Files** – Object replication is much more efficient for moving large files and has a much stronger ability to replicate data near the throughput limits of the server it is running on.
- **Dense Filesystems** – Customers with dense filesystems with millions of small files will see a major benefit for the time to start and complete incremental replication. Even if the change rate is low, object replication will start much faster due to the time difference required for change discovery, and complete faster with its ability to fill the pipeline.
- **Network Speed** – If the customer is already saturating their lower speed network links using file replication, then they may not see any gains by moving to object replication (aside from the time to start incremental replication depending on filesystem density). However, customers with high speed network links will see an improvement by using object replication, which has been benchmarked to run at speeds of 600MB/s to 900MB/s depending on the test system / filesystem under test.

Specific Features of JetMirror

- **High Speed Data Transfer** – The testing results below are for full replication between the source and target filesystem and compare the difference between file and object replication. The expectation is that customers will see comparable performance improvements in these environments.
- **Quick Detection of Incremental Changes** – The improvement to incremental replication performance is where most customers will see the biggest impact of switching to object replication. Because object replication is able to quickly determine the object changes by comparing the snapshot bitmaps between the source and target filesystems, change detection takes milliseconds rather than minutes, allowing replication to begin immediately. Because all object changes are known immediately, it also ensures that the data pipeline remains full for the more efficient replication. File replication may have wait time while the next changed file is found and processed, which can extend replication times to hours or even days for large, dense filesystems with 100s of millions of files.
- **Replication State Stored on Filesystem** - The replication state (aka Persona Object) is stored on both the source and target filesystem as part of the replication snapshot. This enables greater consistency between the source and target filesystem replication status. More importantly, it allows the replication state to be quickly re-establishing using incremental replication after the replication relationship has been broken. A key use case for this is in the initial deployment of a disaster recovery system. Many times, the WAN link between the primary and DR site is slow, which makes an initial full replication problematic in terms of time and bandwidth requirements. This feature now enables the end user to deploy the DR system at the primary site, connected via the high speed internal LAN for the initial replication. Once the replication has completed, the DR system can be powered down and shipped to the DR site. Once powered on, the replication relationship can be re-established, and the replication state updated with an incremental replication job.
- **More snapshots on target than source** – With object replication, the source and target filesystem can be different sizes, which allows the target filesystem to retain snapshots on the target filesystem longer than on the primary filesystem. This greatly increases the ability to use the DR system as a longer term archive for disaster recovery options.
- **Multi-Target Replication (A to B/C/D)** – A source filesystem can be included in multiple replication schedules, allowing for multi-target replication. The snapshots / replication states for each replication job are separately maintained.
- **Maintain Quotas Across Replication** – As with filesystem / directory structures, quotas are stored as objects. These objects are replicated with the filesystem.
- **Replicating Access Points** – Access points (whether CIFS share or NFS exports) are contained in the registry object which is stored in a special location on filesystem. The registry object is included in the replication job, allowing for enhanced DR for failover and failback. This is discussed further as part of disaster recovery.

H. Storage Virtualization

1. Cluster Namespace and Mixed-Mode host support

An individual or clustered set of BlueArc servers can present a single, unified or “global” namespace to hosts. This unified namespace allows any host accessing any BlueArc clustered server to see the same directory structure. The BlueArc term for this capability is called Cluster Namespace (CNS). When implementing CNS each BlueArc server still owns its own filesystems. When data is requested from a host to a given BlueArc server in the cluster that does not own the filesystem in question, that BlueArc server transfers the request to the appropriate server in the cluster.

Beyond host redirection, the principal advantage of using CNS is simplified storage management at scale. CNS reduces the number of mount points and presents an abstraction layer of the individual filesystems to the end-user or application, thereby giving the storage administrator the freedom to leverage any filesystem within the BlueArc cluster for presentation as a single directory with a hierarchical tree structure. This hierarchy allows for any number of storage tiers to appear as a single filesystem to hosts, with a single mount point if desired. The storage administrator achieves enormous flexibility with CNS; end-users or applications do not necessarily know the type of physical storage on which their data resides. This abstraction allows administrators to best match the type of storage to the classification of data on each tier, without requiring users or applications to know the physical location of the different filesystems. All tiers appear as a single large filesystem, incorporating whichever storage technologies are best suited, scaling to petabytes. CNS also allows administrators the freedom to expand or change the underlying storage architecture without having to notify users or re-write applications.

In enterprise consolidation scenarios, CNS provides a centralized file server structure. Windows, UNIX, Linux, or other software-based file servers have much lower limits on the size of individual filesystems (which is the predominant reason why enterprises have experienced such a proliferation of such systems and now need to consolidate), and have lower performance characteristics as well. With CNS storage administrators can scale to massive capacities and provide a virtual CNS tree for all end-users and applications. As the number of users, projects, and overall data capacity increases, more pseudo-directories can be added within the existing CNS structure and immediately be accessed by the appropriate users. Increases in capacity, number of filesystems, and/or aggregate performance requirements are all easily accommodated with SiliconFS without changes to the way users mount the data exports. In conjunction with BlueArc's virtual storage pool capabilities, CNS enables administrators to configure filesystems to automatically grow as needed within pre-defined rules, eliminating downtime associated with storage provisioning. The end-user view of the file structure never changes as it dynamically scales to meet demands. The ability to access any data, anywhere within a BlueArc CNS structure reduces end-user confusion, eliminates the need to rewrite applications, lowers administrator overhead, and provides a flexible, robust design to better accommodate the storage needs of both today and tomorrow.

2. EVS and Secure EVS support

As discussed previously, Enterprise Virtual Servers and Secure EVS partitioning are two important storage virtualization features. The EVS concept is central to SiliconFS's high-availability design, and provides the storage administrator with a powerful and convenient method to perform maintenance, load balancing, and data migration functions without compromising system uptime.

Virtual Servers may reside within one security domain (i.e., one user authentication framework) or across multiple security domains allowing for separated, partitioned, secure operation. The storage administrator does not have to sacrifice centralized storage and centralized management, but can still make the BlueArc cluster look like many independent servers, each possibly with its own security model. Such a configuration is very useful in academic research environments for example, where it is common to have multiple departments sharing centralized resources, yet each wanting to maintain their own independent security domains. Multi-tenant environments for hosted data services (as typically offered by datacenter co-location facilities) also have the same requirements and would enjoy the same benefits.

When configuring Virtual Servers, there are a few common resources which are defined at the physical server level that each EVS inherits; e.g., physical server DNS and Domain Name entries. Aside from these shared parameters, each EVS can be individually tailored with additional parameters including CIFS shares and/or NFS exports, IP addresses, and an independent host name.

3. BlueArc Virtualization framework

BlueArc provides a sophisticated storage virtualization framework, intended to free the storage administrator from mundane storage architecture duties, simply management of SiliconFS, and enable dynamic expansion of the total data capacity under management. The underlying concept of the virtualization framework is the concept of Storage Pools, an abstraction object to which storage administrators assign various properties, like maximum size for example.

Storage Pools are created from one or more System Drives (SDs), a BlueArc term for what most readers will identify as Logical Unit Numbers, or LUNs, used in SCSI terminology to identify the physical target for storage operations. Preferably, Storage Pools contain a large number of SDs, spread out over a large number of physical RAID controllers. The greater the number and spread of SDs the better the performance of the entire Storage Pool, and the greater the initial data capacity may be.

Storage Pools provide a layer of virtualization over the SD itself; the most immediate and obvious advantage of this arrangement is that the Storage Pool can be sized independently of any single SD. This virtualization frees the storage administrator from having to plan LUN sizing ahead of time, or from having to copy data off particular LUNs in order to resize or otherwise redefine an existing RAID array consisting of some number of LUNs. Such limitations plagued early SAN implementations, as storage administrators could not know ahead of time how large a particular RAID set might need to grow, nor how many or what type LUNs might be needed (to say nothing of migrating data from older disk hardware to new as the infrastructure aged....)

Filesystems are defined within Storage Pools in SiliconFS, and a given Storage Pool may have a large number of filesystems. Storage administrators create filesystems based on individual application, user, group, or other business needs, but the raw filesystem is not itself exposed to the end-user or application. Filesystems are individually bound to Enterprise Virtual Servers – it is at the EVS layer that network exports are created (with dependencies on IP addresses, for example). The host computer sees the BlueArc “server” as that EVS (with the specified IP address) with whatever network exports are defined. In this way a single EVS may contain a number of network exports, each of which specifies a filesystem underneath. When using the BlueArc Cluster Namespace feature, a number of filesystems may be organized into a single tree structure, with the root of the tree exported to hosts, so that the entire collection looks like a single filesystem.

There are a number of advantages to filesystem and Storage Pool virtualization. Rebinding of a single filesystem to a different EVS to achieve fine-grained control over load-balancing without needing to reconfigure the underlying storage, for example. Taking a snapshot of a single filesystem, or rolling back an entire filesystem to any given consistent checkpoint is another. Flexibility in sizing filesystems is a very useful advantage. Perhaps only a small amount of data needs to be stored, but paying the performance penalty of using a small number of disks is not a smart design, so why not spread the small amount of data in one small filesystem over as large a Storage Pool as possible?

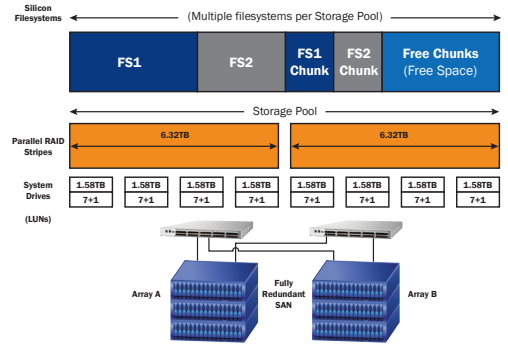


Figure 2: Schematic representation of SiliconFS with Storage Pools

Virtual Volumes (ViVols) are another component of the BlueArc Virtualization Framework, but exist apart from Storage Pools. The concept and function of a Virtual Volume is the creation of a logical container that allows for quotas to be applied to whatever filesystems reside within that container. Because the ViVol is a logical object, as long as there is space available in the underlying storage the ViVol can be expanded (or contracted) automatically. Likewise, if a ViVol is contracted or deleted the disk space consumed by it is returned to the underlying storage, freeing up space which other ViVols may use.

Because ViVols have properties similar to that of any physical volume, the storage administrator may easily manage and monitor various ViVols as separate entities. Yet ViVols are an administrative tool; end-users and applications have no concept or knowledge that they exist. ViVols can be created anywhere on the underlying storage and may be dynamically expanded or contracted at any time by adjusting the quota associated with them. SiliconFS supports quota definitions on a per-user, -group, and/or -ViVol basis.

Quotas allow the administrator to define the amount of space that a volume, Virtual Volume, group or user may consume. If a disk-based quota is set, the system reports the available space available to the user or group, based on the defined quota limit. Disk-based quotas may be defined in bytes, kilobytes, megabytes or gigabytes.¹³ All quota settings (Volume, Virtual Volume, user and group quotas) have the same quota properties, namely:

- Quota limit
- Hard or soft limit
- Warning threshold
- Critical threshold
- e-mail recipient alert event list

Quota event generation is designed to provide the storage administrator with prompt notification without an inundation of event messages. The warning message is sent when the quota threshold is first exceeded. No additional messages are sent until the initial problem has been resolved and a subsequent quota violation occurs. This setup avoid the hysteresis effect when volumes fluctuate rapidly just below or just above a given quota threshold.

I. BlueArc Open Storage philosophy

A large amount of the hardware platform used by SiliconFS is proprietary, and must be almost by definition. Even if SiliconFS were open-source, how could others apply it to extend or change the implementation? SiliconFS must be executed largely in FPGAs and cannot be placed “on top of” normal host operating systems, unlike many other open-source filesystem offerings. A deep knowledge of VLSI programming is necessary for any software architect to understand how to implement even basic functions of SiliconFS.

For these reasons BlueArc is actively developing the BlueArc Data Management Framework, a set of open software API's for which others may more readily program additional features. An example of this extensibility is the use of the Data Migrator feature for third-party enterprise storage products, like Hitachi's Data Discovery Suite, or the integration of specific management software suites (e.g., Hitachi's HiCommand) for specific RAID controller manufacturers.

Beyond the BlueArc Data Management Framework, BlueArc maintains an aggressively open storage and platform philosophy. Unlike other filesystem vendors BlueArc is not in the business of developing or selling proprietary software for host-side connectivity. Vendors of proprietary software protocols assert that anything open must be slow, clumsy, or somehow ill-suited to the tasks at hand (NFS is a typical example), but their view overlooks the benefits of ubiquity and standardization in the community, not to mention the avoidance of vendor lock-in. BlueArc

¹³. This is the current configuration, future versions may allow quotas to be defined in larger increments.



instead prefers to rely on well-established, agreed-upon industry standards (preferably standards which do not change often). For this reason BlueArc helps develop and fully implements and supports protocols such as NFS, iSCSI, NDMP, etc. BlueArc would rather redesign the file server itself (for higher performance, larger scalability, advanced features, etc.) and keep the open protocols in place, rather than scrap the protocols and go the proprietary route.

The BlueArc Open Storage philosophy extends to other areas outside the filesystem as well, e.g., authentication frameworks and choice of back-end RAID controller manufacturers. Better to use open standards such as LDAP, Active Directory, or NIS/NIS+ than create another vendor-specific, closed solution. Better to offer the customer a range of well-supported RAID controller technologies to better architect specific storage tiers for specific business needs, rather than force-fit a limited range of rebranded, OEM'ed, or otherwise proprietary technologies. The storage industry is already plagued by a number of closed storage manufacturers, each seeking to lock the customer in to their specific technology solutions. This direction is not in agreement with the BlueArc Open Storage philosophy. There have been many attempts to market what “open” storage really means. For BlueArc being “open” means working with as many protocols, frameworks, and technology manufacturers as feasible, a strict adherence to industry standards, and having the company's future product direction driven by the voice of the customer. BlueArc cannot support every protocol, framework, or manufacturer out there (even the industry standard ones) but we can support the greatest number of them that satisfy the largest amount of customer requirements.

J. Future-proofing

The filesystem universe is littered with implementations which address narrow segments of the market. Distributed network filesystems are ubiquitous, providing persistent storage duties for simultaneous host access, and useful enterprise data management features as well, but are usually characterized by poor performance and limited scale. Shared-SAN filesystems are designed to open SAN architectures to network hosts, but suffer from lack of metadata scalability and are expensive to implement across the enterprise. Parallel filesystems offer tremendous storage bandwidth, very high scalability, and have built-in availability features, but are largely proprietary and extremely complex to implement and continuously tune for optimal performance. Achieving a balance between high performance, high scalability, enterprise data management features, and use of ubiquitous network filesystem protocols requires a “round” filesystem platform.

SiliconFS is round in the sense that it is a filesystem which provides high scalability, fast metadata processing performance, and excellent data movement speed under a wide range of host loads, usage patterns, and data types. Other filesystems are more like “point” solutions, optimized either for performance as defined by narrow criteria, or for specific features or applications. As such, these filesystems were designed to perform well only under certain loads, access patterns, and data types. Much of the marketing behind these filesystems is geared to defining the limited range where those filesystems perform well, and hiding from the broader, non-optimal situations. The BlueArc philosophy for SiliconFS is different in that it seeks to be useful across the widest range of loads, access patterns, and data types. Not specifically designed for any corner cases, SiliconFS offers the storage administrator filesystem flexibility and a degree of future-proofing: even unforeseen requirements can be handled with minimal effort or re-architecting of existing solutions.

K. Conclusions

BlueArc is proud to build robust, flexible, advanced storage solutions. The BlueArc System Software with SiliconFS at its core is the foundation for one of the most open, most adaptable, most future-proof network storage solutions available.



BLUEARC SILICONFS

Implementing BlueArc storage products has many key benefits for our enterprise customers:

- simplified, extremely easy-to-use data management
- a high degree of data protection for business continuity
- transparent data mobility for any number of different storage tiers
- industry-leading scalability
- exceptional performance
- low total cost-of-ownership

These attributes contribute to BlueArc's strong presence in many industry segments. Worldwide, customers on the forefront of their respective industries rely on BlueArc solutions for their critical application needs. These customers have reaped the benefits of SiliconFS to better serve the needs of their users, now and into the future.

With the SiliconFS architecture BlueArc has embarked on a journey which leverages its numerous successful deployments with customers in many markets and its leading performance and scalability to enable:

- Enhanced metadata scalability for high-performance computing (HPC) environments
- Outstanding single-server IOPS performance
- High storage throughput using open, standardized network filesystem protocols
- A platform for unified storage management: scalable and predictable performance, unmatched scalability, and sophisticated data management functions in a single platform

While there may be other storage solutions for specific market segments, the BlueArc System Software with SiliconFS delivers exceptional functionality across a broad range of customer requirements and thus has the largest applicability in the world of filesystems available today. SiliconFS delivers the performance and scalability benefits of a SAN with the ease of management and client neutrality of NAS, all in one unified solution. Although our hardware architecture is unique, SiliconFS uses standard disk architectures, standard network protocols, standard management protocols, and standard backup protocols. BlueArc strongly believes in client neutrality, not in tying our customers to any vendor-specific solution. Ubiquity is the lingua franca of our storage solutions – we architect our platform to serve our customers' needs.

About BlueArc

BlueArc is a leading provider of high performance unified network storage systems to enterprise markets, as well as data intensive markets, such as electronic discovery, entertainment, federal government, higher education, Internet services, oil and gas and life sciences. Our products support both network attached storage, or NAS, and storage area network, or SAN, services on a converged network storage platform.

We enable companies to expand the ways they explore, discover, research, create, process and innovate in data-intensive environments. Our products replace complex and performance-limited products with high performance, scalable and easy to use systems capable of handling the most data intensive applications and environments. Further, we believe that our energy efficient design and our products' ability to consolidate legacy storage infrastructures, dramatically increases storage utilization rates and reduces our customers' total cost of ownership.



BlueArc Corporation
Corporate Headquarters
50 Rio Robles
San Jose, CA 95134
t 408 576 6600
f 408 576 6601
www.bluearc.com

BlueArc UK Ltd.
European Headquarters
Queensgate House
Cookham Road
Bracknell RG12 1RB, United Kingdom
t +44 (0) 1344 408 200
f +44 (0) 1344 408 202