Abstract

EMC ScaleIO is software that creates a server SAN, aggregating servers’ local direct-attached storage to form globally accessible shared block storage. This white paper describes the functionality of ScaleIO and explains the converged architecture, scalability capabilities, elasticity features and the massively parallel, linearly scalable performance of the system.

December 2013
**Table of Contents**

**Introduction** ........................................................................................................ 4

**EMC ScaleIO** ........................................................................................................ 6

**Architecture** ......................................................................................................... 6
- Software Components.......................................................................................... 6
- Convergence of Storage and Compute ............................................................... 7
- Pure Block Storage Implementation ................................................................... 7
- Massively Parallel, Scale-Out I/O Architecture ............................................... 8
- Mix-and-Match Nodes ....................................................................................... 8
- Hardware Agnostic ............................................................................................. 8
- Volume Mapping and Volume Sharing .............................................................. 8
- Clustered, Striped Volume Layout ..................................................................... 9

**Resilience and Elasticity** .................................................................................... 9
- Redundancy Scheme and Rebuild Process ........................................................ 9
- Elasticity and Rebalancing ................................................................................ 10
- Software-Only—But As Resilient As A Hardware Array .................................. 11
- Managing Clusters of Hundreds or Thousands of Nodes ............................... 11
- Protection Domains ........................................................................................... 12

**Advanced Functionality** ..................................................................................... 13
- IOPS Limiter ........................................................................................................ 13
- Encryption at Rest ............................................................................................. 13
- Snapshots ........................................................................................................... 13
**Introduction**

As enterprises consolidate into mega data centers and small-to-medium-sized businesses move to cloud and hosting infrastructures, data centers are rapidly expanding to thousands of servers. A recent survey revealed the top 10 concerns and challenges of data center operators:

![Figure 1: Top 10 Challenges of Data Center Operators](image)

Management, capacity planning, power availability, and costs are among the main challenges that operators of modern data centers face. Data storage is the origin of many of these challenges.

Storage solutions are often divided into three categories: object storage, file storage (NAS), and block storage (SAN). The market is flooded with commercial and open-source object and file storage today, but despite the $20 billion market for SAN solutions, only a limited number of block-storage vendors can address the unrelenting demand for robust, large-scale block storage.

Traditional SAN storage offers the high performance and high availability required to support business applications, hypervisors, file systems, and databases. However, despite widespread acceptance, traditional SAN solutions come with many inherent challenges. A SAN does not address a data center’s need for linear scalability and high performance at a low total cost of ownership (TCO). Costly and complex to implement and operate, SAN storage requires host bus adapter (HBA) cards on the application servers, fabric switches, and storage arrays that demand considerable planning, complex configuration, and lengthy implementation cycles.
Unlike server hardware, which is open and commoditized, the SAN market is still captured by vendor lock-in, which results in high costs for customers and barriers to interoperability. Although direct-attached storage (DAS) and SAN storage use the same types of disks, disks in SAN arrays generally cost many times more than disks in servers. And because traditional SAN systems are so complex to size and scale, customers often end up buying an oversized SAN configuration. The result is a capital expenditure that renders storage costs an unwarranted portion of the IT budget.

Ironically, the performance of traditional SAN systems goes down as more servers are added, because all the servers compete for the same centralized storage system’s CPU, cache, and disk I/O resources. Eventually, configuration changes and SAN additions are required, increasing the complexity and cost of the infrastructure as well as the accompanying operational risks. Another serious challenge posed by a traditional SAN is finding the skilled staff to run the system. Because only vendor-certified administrators are eligible for such work, they are in high demand and scarce. This scarcity is an unmanageable risk.

Overcoming modern data center challenges calls for a completely new breed of data storage: elastic converged storage. Products designed according to the elastic converged storage paradigm help data center operators successfully manage many of the major challenges. EMC ScaleIO™ enables data centers to cut storage costs by over 80% and streamlines storage operations by converging storage into compute. ScaleIO addresses most of the top 10 concerns and challenges expressed by data center operators.
**EMC ScaleIO**

ScaleIO is a software-only solution that uses existing hosts’ local disks and LAN to realize a virtual SAN that has all the benefits of external storage—but at a fraction of the cost and the complexity. ECS turns existing local internal storage into internal shared block storage that is comparable to or better than the more expensive external shared block storage. The lightweight ScaleIO software components are installed in the application hosts and inter-communicate via a standard LAN to handle the application I/O requests sent to ScaleIO block volumes. An extremely efficient decentralized block I/O flow combined with a distributed, sliced volume layout results in a massively parallel I/O system that can scale to hundreds and thousands of nodes.

ScaleIO was designed and implemented with enterprise-grade resilience as a must-have. Furthermore, the software features efficient distributed auto-healing processes that overcome media and node failures without requiring administrator involvement. Dynamic and elastic, ScaleIO enables administrators to add or remove nodes and capacity “on the fly.” The software immediately responds to the changes, rebalancing the storage distribution and achieving a layout that optimally suits the new configuration. Because ScaleIO is hardware agnostic, the software works efficiently with various types of storage, including magnetic disks, solid-state disks, and PCIe flash cards, networks, and hosts. It can be easily installed in an existing infrastructure as well as in greenfield configurations.

**Architecture**

**Software Components**

The ScaleIO Data Client (SDC) is a lightweight device driver situated in each host whose applications or file system requires access to the ScaleIO virtual SAN block devices. The SDC exposes block devices representing the ScaleIO volumes that are currently mapped to that host.

The ScaleIO Data Server (SDS) is a lightweight software component that is situated in each host that contributes local storage to the central ScaleIO virtual SAN.
Convergence of Storage and Compute

ScaleIO converges the storage and application layers. The hosts that run applications can also be used to realize shared storage, yielding a wall-to-wall, single layer of hosts. Because the same hosts run applications and provide storage for the virtual SAN, an SDC and SDS are typically both installed in each of the participating hosts. Another option is to combine nodes that have only SDCs installed and nodes that have only SDSs installed.

Carefully designed and implemented to consume the minimum computing resources required for operation, the ScaleIO software components have a negligible impact on the applications running in the hosts. For example, one can build a more traditional two-layer system in which the upper-layer application hosts access a virtual, software-only SAN, whose storage resources are embedded inside a second layer of hosts. The latter layer acts as a scale-out, software-defined storage “array.” In such a configuration, SDCs are installed in the upper-layer hosts, and SDSs run in the lower-layer hosts.

Pure Block Storage Implementation

ScaleIO implements a pure block storage layout. Its entire architecture and data path are optimized for block storage access needs. For example, when an application submits a read I/O request to its SDC, the SDC instantly deduces which SDS is responsible for the specified volume address and then interacts directly with the relevant SDS. The SDS reads the data (by issuing a single read I/O request to its local storage or by just fetching the data from the cache in a cache-hit scenario), and returns the result to the SDC. The SDC provides the read data to the application.

This flow is very simple, consuming as few resources as necessary. The data moves over the network exactly once, and a maximum of only one I/O request is sent to the SDS storage. The write I/O flow is similarly simple and efficient. Unlike some block storage systems that run on top of a file system or object storage that runs on top of a local file system, ScaleIO offers optimal I/O efficiency.
Massively Parallel, Scale-Out I/O Architecture

ScaleIO can scale to hundreds or even thousands of nodes, thus breaking the traditional scalability barrier of block storage. Because the SDCs propagate the I/O requests directly to the pertinent SDSs, there is no central point through which the requests move—and thus a potential bottleneck is avoided. This decentralized data flow is key to the linearly scalable performance of ScaleIO. Therefore, a large ScaleIO configuration results in a massively parallel system. The more servers or disks the system has, the greater the number of parallel channels that will be available for I/O traffic and the higher the aggregated I/O bandwidth and IOPS will be.

Mix-and-Match Nodes

The vast majority of traditional scale-out systems are based on a “symmetric brick” architecture. Unfortunately, medium and large data centers cannot be standardized on exactly the same bricks for a prolonged period, because hardware configurations and capabilities change over time. Therefore, such symmetric scale-out architectures are bound to run in small islands. ScaleIO was designed from the ground up to support a mix of new and old nodes with dissimilar configurations.

Hardware Agnostic

ScaleIO is platform agnostic and works with existing underlying hardware resources. Besides its compatibility with various types of disks, networks, and hosts, it can take advantage of the write buffer of existing local RAID controller cards—and can also run in servers that do not have a local RAID controller card.

For the local storage of an SDS, one can use internal disks, directly attached external disks, virtual disks exposed by an internal RAID controller, partitions within such disks, and more. Partitions can be useful to combine system boot partitions with ScaleIO capacity on the same raw disks. If the system already has a large, mostly unused partition, ScaleIO does not require repartitioning of the disk, as the SDS can actually use a file within that partition as its storage space.

Volume Mapping and Volume Sharing

The volumes that ScaleIO exposes to the application clients can be mapped to one or more clients running in different hosts. Mapping can be changed dynamically if necessary. In other words, ScaleIO volumes can be used by applications that expect shared-everything block access (such as Oracle RAC) and by applications that expect shared-nothing or shared-nothing-with-failover access.
Clustered, Striped Volume Layout

A ScaleIO volume is a block device that is exposed to one or more hosts. It is the equivalent of a logical unit in the SCSI world. ScaleIO breaks each volume into a large number of data chunks, which are scattered across the SDS cluster’s nodes and disks in a fully balanced manner. This layout practically eliminates hot spots across the cluster and allows for the scaling of the overall I/O performance of the system through the addition of nodes or disks. Furthermore, this layout enables a single application that is accessing a single volume to use the full IOPS of all the cluster’s disks. This flexible, dynamic allocation of shared performance resources is one of the major advantages of converged scale-out storage.

Resilience and Elasticity

Redundancy Scheme and Rebuild Process

ScaleIO uses a mirroring scheme to protect data against disk and node failures. The ScaleIO architecture supports a distributed two-copy redundancy scheme. When an SDS node or SDS disk fails, applications can continue to access ScaleIO volumes; their data is still available through the remaining mirrors. ScaleIO immediately starts a seamless rebuild process whose goal is to create another mirror for the data chunks that were lost in the failure. In the rebuild process, those data chunks are copied to free areas across the SDS cluster, so it is not necessary to add any capacity to the system. All the surviving SDS cluster nodes together carry out the rebuild process by using the aggregated disk and network bandwidth of the cluster. As a result, the process is dramatically faster—resulting in a shorter exposure time and less application-performance degradation. On the completion of the rebuild, all the data is fully mirrored and healthy again. If a failed node rejoins the cluster before the rebuild process has been completed, ScaleIO dynamically uses the rejoined node’s data to further minimize the exposure time and the use of resources. This capability is particularly important for overcoming short outages efficiently.
Elasticity and Rebalancing

Unlike many other systems, a ScaleIO cluster is extremely elastic. Administrators can add and remove capacity and nodes “on the fly” during I/O operations. When a cluster is expanded with new capacity (such as new SDSs or new disks added to existing SDSs), ScaleIO immediately responds to the event and rebalances the storage by seamlessly migrating data chunks from the existing SDSs to the new SDSs or disks. Such a migration does not affect the applications, which continue to access the data stored in the migrating chunks. By the end of the rebalancing process, all the ScaleIO volumes have been spread across all the SDSs and disks, including the newly added ones, in an optimally balanced manner. Thus, adding SDSs or disks not only increases the available capacity but also increases the performance of the applications as they access their volumes.

![Diagram](image1)

**Figure 3: Automatic rebalancing when disks are added**

When an administrator decreases capacity (for example, by removing SDSs or removing disks from SDSs), ScaleIO performs a seamless migration that rebalances the data across the remaining SDSs and disks in the cluster.

![Diagram](image2)

**Figure 4: Automatic rebalancing when disks are removed**

Note that in all types of rebalancing, ScaleIO migrates the least amount of data possible. Furthermore, ScaleIO is flexible enough to accept new requests to add or remove capacity while still rebalancing previous capacity additions and removals.
**Software-Only—But As Resilient As A Hardware Array**

Traditional storage systems typically combine system software with commodity hardware—which is comparable to application servers’ hardware—to provide enterprise-grade resilience. With its contemporary architecture, ScaleIO provides similar enterprise-grade, no-compromise resilience by running the storage software directly on the application servers. Designed for extensive fault tolerance and high availability, ScaleIO handles all types of failures, including failures of media, connectivity, and nodes, software interruptions, and more. No single point of failure can interrupt the ScaleIO I/O service. In many cases, ScaleIO can overcome multiple points of failure as well.

**Managing Clusters of Hundreds or Thousands of Nodes**

Many storage cluster designs use tightly coupled techniques that may be adequate for a small number of nodes but begin to break when the cluster is bigger than a few dozen nodes. The loosely coupled clustering management schemes of ScaleIO provide exceptionally reliable—yet lightweight—failure and failover handling in both small and very large clusters.

Most clustering environments assume exclusive ownership of the cluster nodes and may even physically fence or shut down malfunctioning nodes. ScaleIO uses application hosts. The ScaleIO clustering algorithms are designed to work efficiently and reliably without interfering with the applications with which ScaleIO coexists. ScaleIO will never disconnect or invoke IPMI shutdowns of malfunctioning nodes, because they may still be running healthy applications.
Protection Domains

A large ScaleIO storage pool can be divided into multiple protection domains, each of which contains a set of SDSs. ScaleIO volumes are assigned to specific protection domains. Protection domains are useful for mitigating the risk of a dual point of failure in a two-copy scheme or a triple point of failure in a three-copy scheme.

![Diagram of Protection Domains](image)

**Figure 5: Protection domains**

For example, if two SDSs that are in different protection domains fail simultaneously, no data will become unavailable. Just as incumbent storage systems can overcome a large number of simultaneous disk failures as long as they don’t occur within the same shelf, ScaleIO can overcome a large number of simultaneous disk or node failures as long as they don’t occur within the same protection domain.
Advanced Functionality

IOPS Limiter

The ScaleIO volume layout enables a single application to use the full IOPS of the entire protection domain’s storage devices. The ability to dynamically allocate and share storage and performance resources is a major advantage of converged scale-out storage. However, sometimes it is preferable to control the behavior of applications that tend to consume more I/O bandwidth or IOPS than desired. Using the ScaleIO IOPS limiter, administrators can set maximum IOPS or bandwidth values per client/volume. If an application attempts to consume more than it is allowed, ScaleIO fluidly limits the application’s IOPS or bandwidth workload.

Encryption at Rest

With ScaleIO, volume data at rest can be stored in an encrypted form, enabling organizations to secure their data while maintaining current service levels for operations.

Snapshots

For each ScaleIO volume, administrators can create dozens of fully rewritable redirect-on-write snapshots. Each snapshot is essentially a volume of its own. The snapshot hierarchy is totally flexible—for example, a snapshot of a snapshot can be created or, if necessary, a volume can be deleted while its snapshots are retained. All the expected restore functionalities are fully supported; one can effortlessly restore a snapshot to its ancestor.

Furthermore, ScaleIO enables one to take a set of consistent snapshots of a given set of volumes across multiple servers. In fact, it is possible to snap the entire cluster’s volumes in a consistent manner. As long as crash consistency is acceptable, there is no need to quiesce any application activities during the creation of snapshots.