

SCALEIO IMPLEMENTATION AND MANAGEMENT

August, 2015



Copyright © 2015 EMC Corporation. All Rights Reserved.

Welcome to ScaleIO 1.30 Design and Implementation Workshop.

Copyright © 1996, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014 EMC Corporation. All Rights Reserved. EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

EMC2, EMC, Data Domain, RSA, EMC Centera, EMC ControlCenter, EMC LifeLine, EMC OnCourse, EMC Proven, EMC Snap, EMC SourceOne, EMC Storage Administrator, Acartus, Access Logix, AdvantEdge, AlphaStor, ApplicationXtender, ArchiveXtender, Atmos, Authentica, Authentic Problems, Automated Resource Manager, AutoStart, AutoSwap, AVALONidm, Avamar, Captiva, Catalog Solution, C-Clip, Celerra, Celerra Replicator, Centera, CenterStage, CentraStar, ClaimPack, ClaimsEditor, CLARiiON, ClientPak, Codebook Correlation Technology, Common Information Model, Configuration Intelligence, Configuresoft, Connectrix, CopyCross, CopyPoint, Dantz, DatabaseXtender, Direct Matrix Architecture, DiskXtender, DiskXtender 2000, Document Sciences, Documentum, elnput, E-Lab, EmailXaminer, EmailXtender, Enginuity, eRoom, Event Explorer, FarPoint, FirstPass, FLARE, FormWare, Geosynchrony, Global File Virtualization, Graphic Visualization, Greenplum, HighRoad, HomeBase, InfoMover, Infoscape, Infra, InputAccel, InputAccel Express, Invista, Ionix, ISIS, Max Retriever, MediaStor, MirrorView, Navisphere, NetWorker, nLayers, OnAlert, OpenScale, PixTools, Powerlink, PowerPath, PowerSnap, QuickScan, Rainfinity, RepliCare, RepliStor, ResourcePak, Retrospect, RSA, the RSA logo, SafeLine, SAN Advisor, SAN Copy, SAN Manager, Smarts, SnapImage, SnapSure, SnapView, SRDF, StorageScope, SupportMate, SymmAPI, SymmEnabler, Symmetrix, Symmetrix DMX, Symmetrix VMAX, TimeFinder, UltraFlex, UltraPoint, UltraScale, Unisphere, VMAX, Vblock, Viewlets, Virtual Matrix, Virtual Matrix Architecture, Virtual Provisioning, VisualSAN, VisualSRM, Voyence, VPLEX, VSAM-Assist, WebXtender, xPpression, xPresso, YottaYotta, the EMC logo, and where information lives, are registered trademarks or trademarks of EMC Corporation in the United States and other countries.

All other trademarks used herein are the property of their respective owners.

© Copyright 2014 EMC Corporation. All rights reserved. Published in the USA.

Course Overview

Description	This Instructor-led course provides an in-depth look at ScaleIO, and includes full instruction and hands-on exercises for the implementation and management of the solution. Also discussed are architecture, planning and design considerations, optimization and troubleshooting.
Audience	This training is targeted those who will be designing, deploying and supporting ScaleIO solutions.
Objectives	Upon successful completion of this course, participants should be able to: • Design and implement a ScaleIO solution in a variety of environments • Manage a ScaleIO cluster In both bare metal and hypervisor environments • Monitor and optimize a ScaleIO cluster

*EMC believes the information in this course is accurate as of its publication date. It is based on pre-GA product information, which is subject to change without notice. For the most current information, see **E-Lab Interoperability Navigator** and product release notes at support.emc.com.*



Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

2

This course will provide comprehensive coverage of all aspects of a ScaleIO solution. Class discussions and hands-on labs will drive the learning experience for topics that include planning and design, installation, implementation and management and monitoring of an end-to-end ScaleIO Block storage solution with integrated components (VMware, alternate hypervisors and OpenStack).

This workshop is targeted at a select cross-functional audience within EMC. It is intended for those who will be involved in positioning, deploying and supporting ScaleIO 1.30.

Welcome

- Instructor introductions
- Student introductions
 - ▶ Name
 - ▶ Role
 - ▶ What is your experience with ScaleIO?
 - ▶ What is your current role?
 - ▶ What are your expectations?



Agenda

Modules

Labs

Day 1

- ScaleIO 1.30 Overview
- ScaleIO Cluster Implementation (Linux/Windows/ESXi)

- Paper solutions design lab
- Install ScaleIO cluster #1 of Linux/Windows hosts
- Expand cluster #1 by adding a host
- Install ScaleIO cluster #2 of VMware hosts

Day 2

- Configuration and Management
- Performance and Troubleshooting

- ScaleIO basic configuration, management and monitoring operations on Linux/Windows/ESXi cluster using GUI, CLI and VSI plug-in
- Rebuild/rebalance operations, throttling mechanisms, READ RAM cache configuration
- ScaleIO failures in various components
- Using dashboard reports, ScaleIO logs and diagnostic commands to perform basic troubleshooting

Day 3

- ScaleIO REST API and OpenStack Integration

- ScaleIO REST API basic operations
- Implement ScaleIO in an OpenStack environment



Module 1: ScaleIO Overview

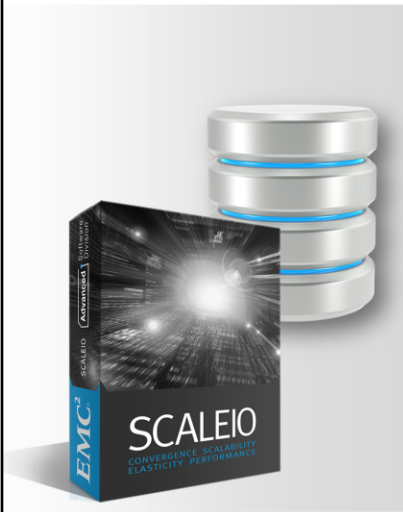
Upon completion of this module, you should be able to:

- Present an overview of the ScaleIO product, architecture and basic internal operation
- Explain the benefits and value
- Perform design of a suitable ScaleIO solution in supported environments



This module provides a high-level technical overview of ScaleIO and explains core concepts relevant to the product. It describes the ScaleIO benefits, customer value proposition and a suitable solution in supported environments.

ScaleIO Overview



- Creates server-based SAN from local storage to deliver performance and capacity on demand
- Creates virtual pool of storage with varying performance tiers
- Hardware-agnostic leveraging any HDDs, SSDs and PCIe Flash cards
- Server-agnostic installing on existing physical and virtual application servers

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

6

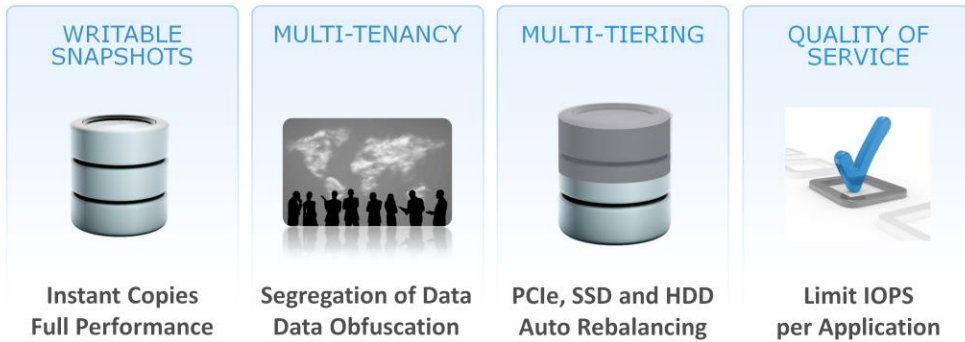
ScaleIO is a software-only solution. It uses the existing local disks on servers and the LAN infrastructure to create a “virtual SAN” with all the benefits of external storage. It uses a sliced distributed volume layout across the participating servers that results in a massively parallel I/O system that can scale to thousands of nodes.

Because ScaleIO is hardware-agnostic, the software works efficiently with various types of disks including: magnetic (HDD) and solid-state disks (SSDs), and Flash PCI Express (PCIe) cards. It can leverage any of these physical media devices to create virtual pools of storage with different user-defined storage tiers.

ScaleIO runs natively on Linux and Windows servers. It is designed to integrate with virtualized host environments, and supports x86-based hypervisors including VMware, Hyper-V, KVM and Xen.

ScaleIO can easily be installed in an existing infrastructure, as well as in green field configurations.

High-end Virtual SAN out of Server-based Storage



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

7

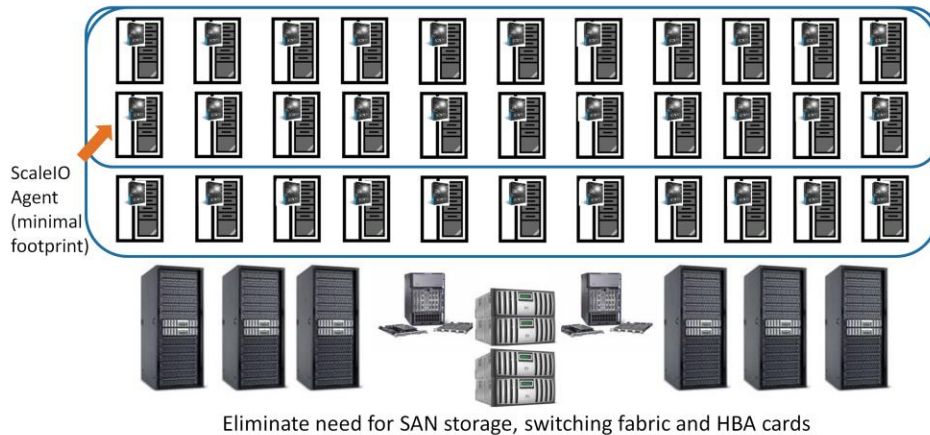
ScaleIO provides for enterprise-class SAN features such as:

- Volume Snapshot - including instant writable snapshots, and consistency groups
- Multi-tenancy - by enabling segregation of tenant data across media and servers and providing data obfuscation
- Multi-tiering - volumes may be tiered, based on type of storage media and class of server and/or networks
- Quality of Service (QoS) configuration - IOPS and bandwidth may be optionally limited to specific values on a per-volume basis

ScaleIO Vision and Value Proposition

Software solution using application servers to create elastic, scalable and resilient virtual SAN that:

- Installs on existing servers that run databases, hypervisors or other apps
- Aggregates application server local disks
- Adds storage and/or compute as needed, as converged solution



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

8

ScaleIO offers a virtual SAN that is entirely server-based.

It is a software solution that installs directly on production servers as a lightweight agent with minimal footprint, enabling them to access aggregated storage from the entire pool of local disks on the servers. Since all the storage in the aggregated pool is server-resident, this removes the need for traditional SAN infrastructure, such as storage arrays and Fibre Channel switches.

The solution can accommodate dynamic growth and shrinkage of the aggregated Storage Pool, without disruption to application I/O activity.

ScaleIO Components and Core Constructs

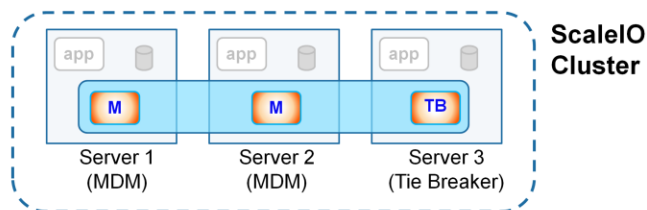
- Software components
 - ▶ Metadata Manager (MDM)
 - ▶ ScaleIO Data Server (SDS)
 - ▶ ScaleIO Data Client (SDC)
- Protection Domains
- Fault Sets
- Storage Pools



These are the software components of ScaleIO and the core constructs within a ScaleIO cluster. We will cover these components in more detail throughout the module.

ScaleIO Metadata Manager (MDM)

- Software-only cluster implemented from scratch
- Shared storage not required
- Replicated repository
- Failover time of ~ 0.5 sec
- Tie-breaker component
- No need for dedicated servers
 - ▶ Dedicated nodes are recommended in very large configurations



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

10

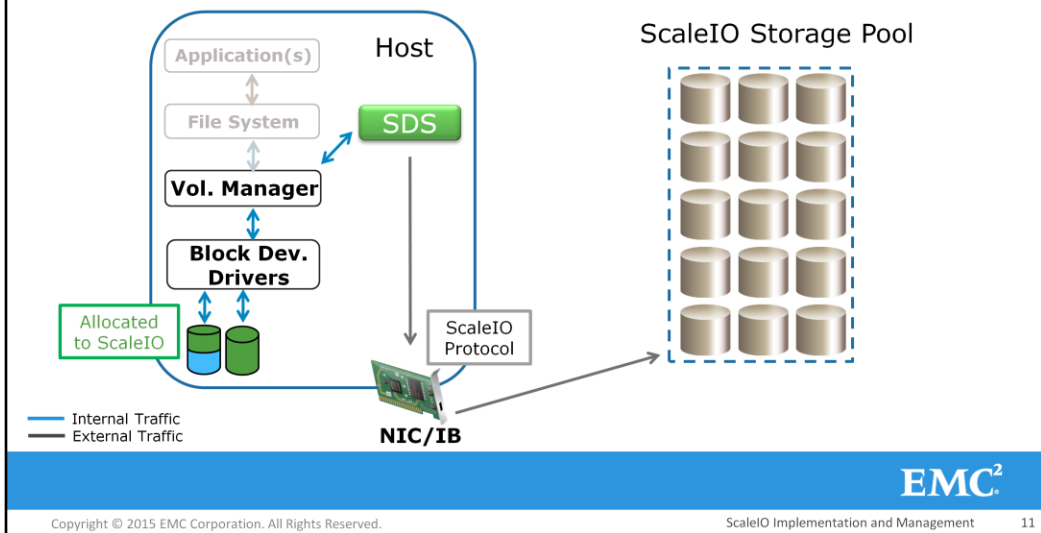
The MDM configures and monitors the ScaleIO system. In production clusters, the MDM should be configured in redundant Cluster Mode, with three members on three servers: Primary MDM, Secondary MDM and Tie-breaker (to handle split-brain scenarios).

The single-mode MDM (using a single server) is a simpler option, but it is not recommended in production deployments.

The MDM cluster aggregates all the physical storage that is configured within ScaleIO to generate a virtual layer, and maintains up-to-date metadata for the entire system.

ScaleIO Data Server (SDS)

- Software daemon/service
- Contributes local storage to ScaleIO Storage Pool

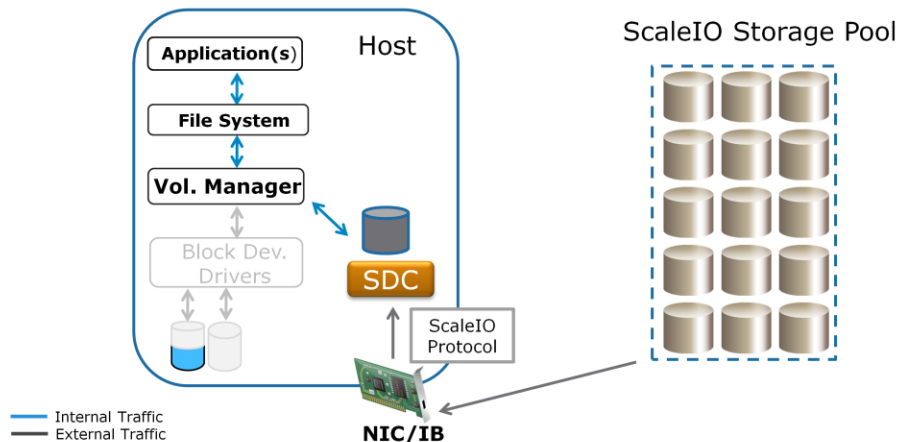


The SDS is a software daemon that enables a server in the ScaleIO cluster to contribute its local storage to the aggregated pool. It manages the capacity of a single server and acts as a backend for data access. The SDS is installed on all servers contributing storage devices to the ScaleIO system. These devices are accessed through the SDS.

Servers can contribute any available local storage for use by ScaleIO. This could be dedicated disks, partitions, or even a file. The “file” option is typically considered only when the customer has already partitioned the disks and is unable to change the partitioning scheme on the production server (for example, a boot disk of the server hosting the operating system on a single partition). In such cases, create a large contiguous file to occupy the unused capacity and allocate the file to ScaleIO.

ScaleIO Data Client (SDC)

- Block device driver
- Exposes ScaleIO shared Block volumes to application
- Looks like any other Block device on OS



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

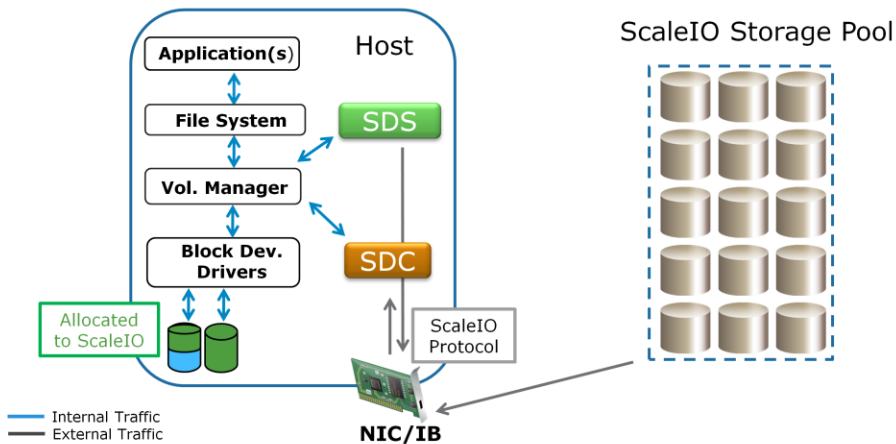
12

The SDC is a lightweight device driver. When installed in a server, it exposes ScaleIO volumes as block devices to any application on the server.

Note that the SDC is a Block device driver that can run alongside other Block device drivers. The only I/Os in the stack that are intercepted by the SDC are those I/Os that are directed at the volume(s) that belong to ScaleIO. There is no interference with I/Os directed at traditional SAN LUNs that are not part of the ScaleIO configuration at the site.

SDS and SDC Co-existence

- SDC and SDS can function on same server
- SDS serves I/O requests of various SDCs
- SDC serves I/O requests of resident host applications



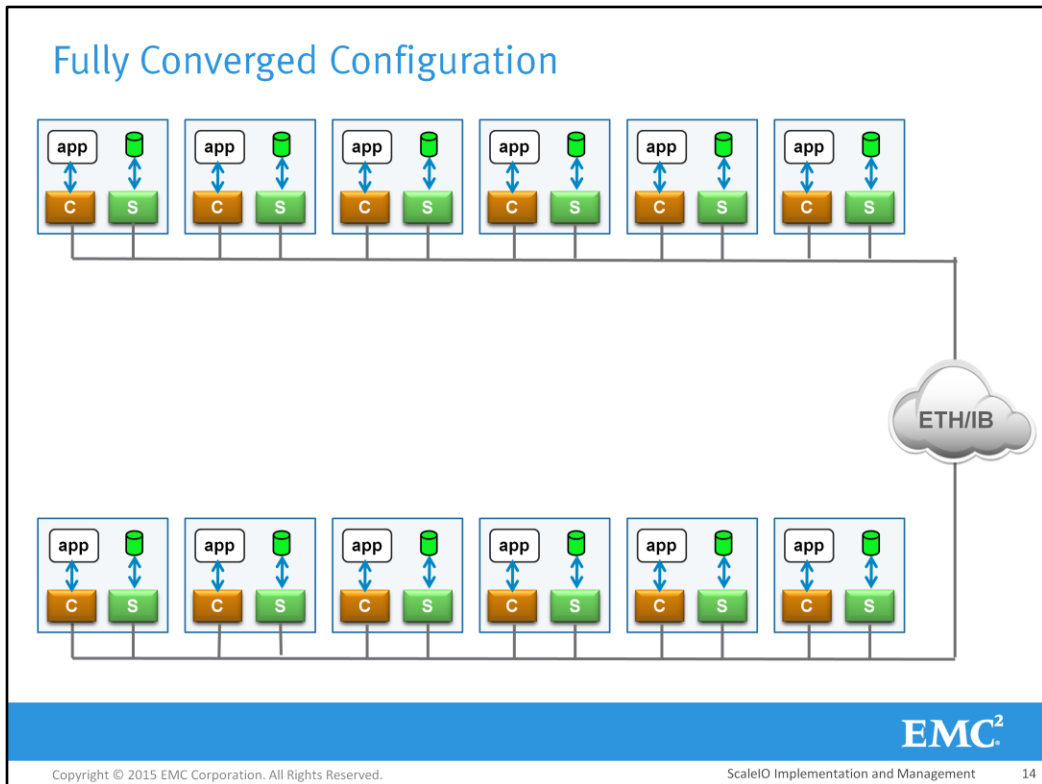
Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

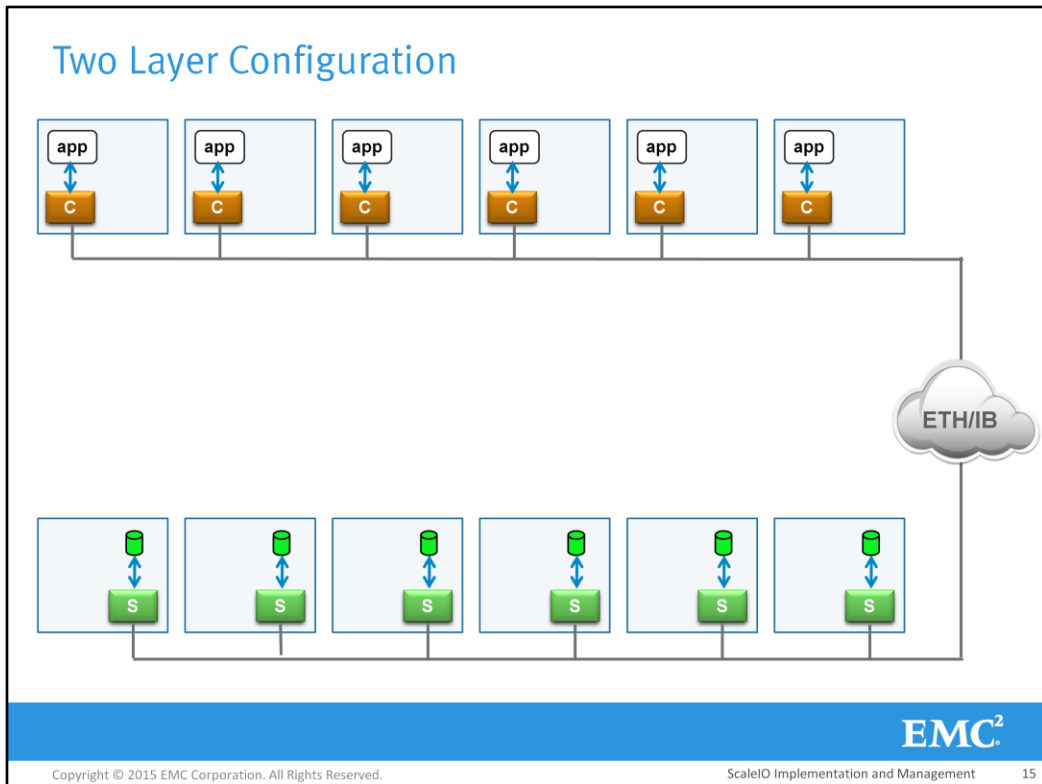
13

EMC²

In typical converged deployments, servers will perform both roles, for example, as a client of ScaleIO consuming Block storage, and as a server contributing storage to ScaleIO. SDC and SDS are both installed on the host to accomplish this.

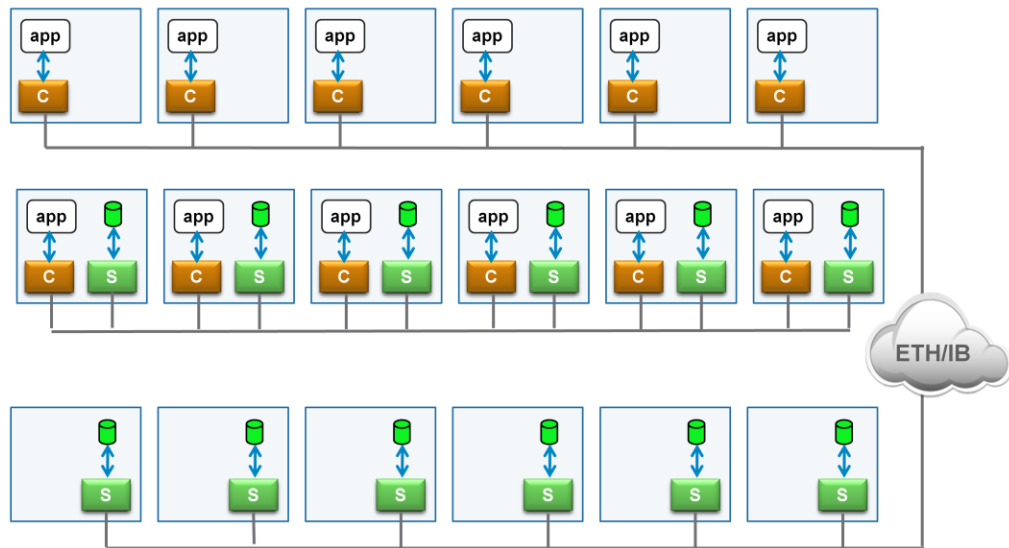


This is the “fully converged” solution with every server performing both roles.



In a two layer deployment, some servers run SDC exclusively and consume storage, while the other servers run SDS exclusively and contribute storage. This emulates the familiar SAN storage architecture. Note in particular that while we get something resembling traditional “SAN-like” centrally managed data storage, it is accomplished without the use of any special hardware such as Fibre Channel HBAs.

Mixed Configuration



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

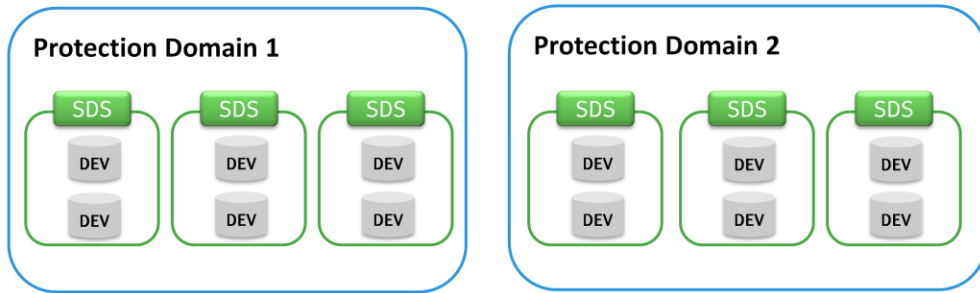
16

Shown here is a mixed configuration that includes some SDC-only, some SDS-only and some dual-role servers.

Note that there is no requirement for “symmetric” nodes - the ScaleIO software can still intelligently determine how to balance the contributed storage over all available spindles to provide optimal performance from the SDCs. Being able to handle asymmetric configurations is a key product differentiator. It is not offered by most competing solutions.

Protection Domains

- Set of SDSs
- Volume defined in Protection Domain
- Maximum 128 SDSs per Protection Domain recommended



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

17

A Protection Domain is a subset of SDSs. Each SDS belongs to one (and only one) Protection Domain. Thus, by definition, each Protection Domain is a unique set of SDSs.

The recommended number of nodes in a Protection Domain is around 128. This yields optimal performance and reduces theoretical MTBF issues.

Criteria for Designing Protection Domains

- Protect against simultaneous failures in large clusters
- Performance isolation
- Data location control (multi-tenancy)
- Network constraints



Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

18

The main reason to create more than one Protection Domain (the default) is to contain the impact of server downtime on storage availability. The key concept is that for any particular ScaleIO volume, data is kept within the boundaries of the Protection Domain from which it is provisioned. With several Protection Domains, one server or device can fail in every Protection Domain, and production I/O will be completely unaffected. This gives the benefit of improved system resilience.

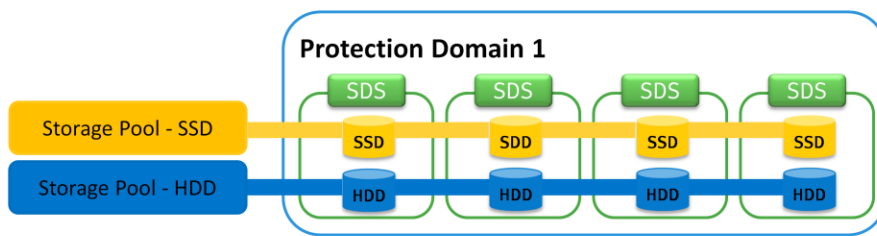
There are other compelling reasons to take advantage of Protection Domains beyond resilience.

- To separate volumes for performance planning, for example, assigning highly accessed volumes in “less busy” domains or dedicating a particular domain to an application, which is essentially server-based tiering
- For data location and partitioning in multi-tenancy deployments, so that tenants can be segregated efficiently and securely
- To accommodate the existing networking infrastructure and layout within the data center

Note that Protection Domains have nothing to do with SDC client access to ScaleIO volumes. An SDC can access all volumes, regardless of Protection Domain assignment. In fact, an SDC can access volumes from several different ScaleIO clusters.

Storage Pools and Tiering

- Storage Pool - set of storage devices in Protection Domain
 - ▶ Volume defined within a Storage Pool
- Benefits
 - ▶ Multi-tiering: Fast vs. slower Storage Pools
 - ▶ Performance-isolation: Multiple Storage Pools of same media speed
 - ▶ Tenant isolation by physical media



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

19

A Storage Pool is a subset of physical storage devices in a Protection Domain. Each storage device belongs to one (and only one) Storage Pool. By default, all devices within a Protection Domain are configured into a single “default” Storage Pool.

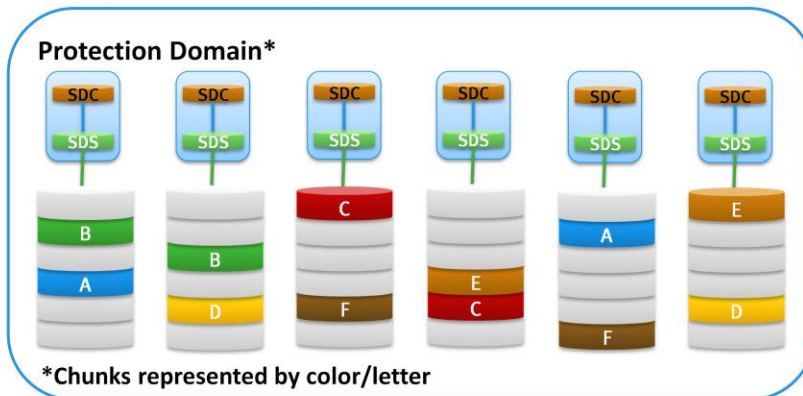
When a ScaleIO volume is configured over the virtualization layer, the volume contents are distributed over all devices residing in the same Storage Pool. Since a Storage Pool can withstand the loss of one of its members, having two failures in two different Storage Pools will not cause data loss.

Storage Pools enable creating different storage tiers in the ScaleIO system. The best practice is to have the same type of storage devices within a Storage Pool. This ensures that each volume will be distributed over the same type of storage devices. Mixing devices of different types within the same storage pool would reduce the performance to that of the slowest device type in the pool.

Beyond tiering, there are other reasons for configuring Storage Pools, including data isolation in multi-tenant environments and performance isolation within a set of devices of the same type.

Data Distribution in ScaleIO Storage Pool

- ScaleIO volumes organized in 1MB chunks replicated across two different servers
- Uses “mesh mirroring” scheme
- Primary and secondary chunks - balanced mix on each SDS disk



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

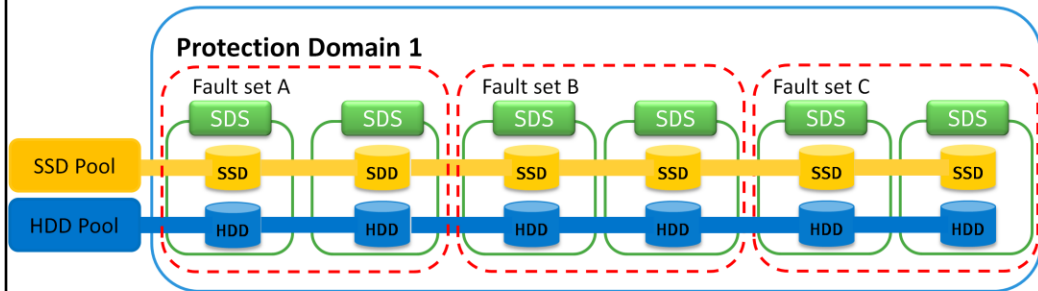
ScaleIO Implementation and Management

20

Internally, every ScaleIO volume is organized into 1MB chunks that are distributed across all devices within the Storage Pool from which the volume is provisioned. ScaleIO maintains mirrored copies of every data chunk in a volume in two different devices in different servers. This ensures resilience with the ability to survive the loss of one device or one server participating in a Storage Pool.

Fault Sets

- Sub-sets of Protection Domain
- Allows grouping SDSs likely to experience a failure together
 - ▶ Main use-case: Rack containing several SDSs



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

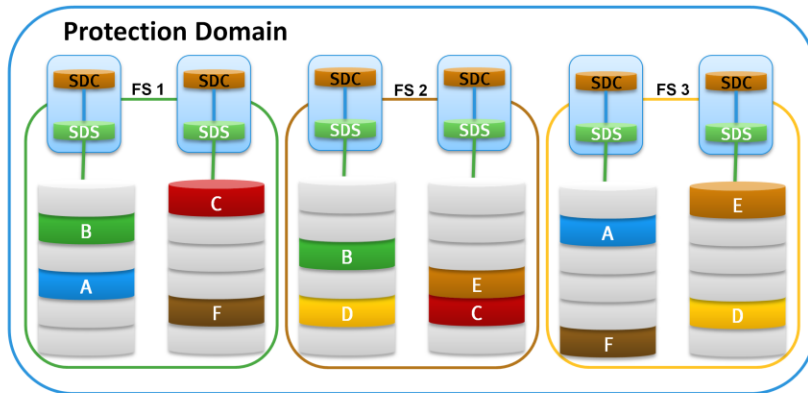
21

A Fault Set is a logical entity that contains a group of SDSs within a Protection Domain. A Fault Set can be optionally defined for a set of servers that are likely to fail together, for example if they are all physically located and powered within the same rack. By design, ScaleIO will never maintain both copies of a given chunk in the same Fault Set. Thus, availability is assured, even if all the servers within one Fault Set fail simultaneously.

Fault Sets should be defined and SDSs added to them at the time of installation. By default, no Fault Sets are created in a Protection Domain.

Data Distribution on Fault Sets

ScaleIO mirrored copies of Block are never in same Fault Set



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

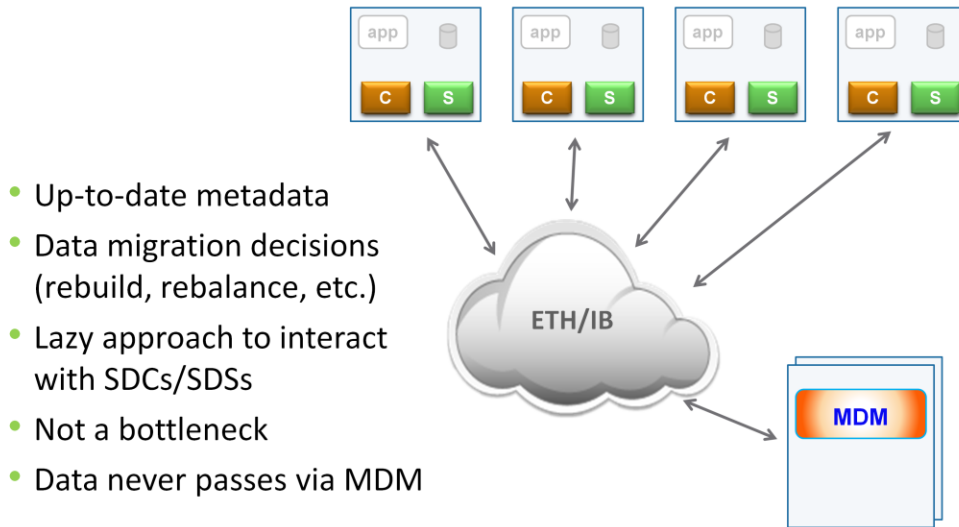
ScaleIO Implementation and Management

22

This is a simple example of Fault Sets at work. Note that each of the chunks (A, B, C, D, E, and F) have two different mirror legs in separate Fault Sets.

You must define at least three Fault Sets, and there must be balanced capacity for each Storage Pool across the different Fault Sets. ScaleIO depends on the user to ensure balanced capacity between the Fault Sets.

ScaleIO MDM - Metadata Handling



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

23

The MDM is the component that orchestrates placement and movement of chunks within an SDS cluster.

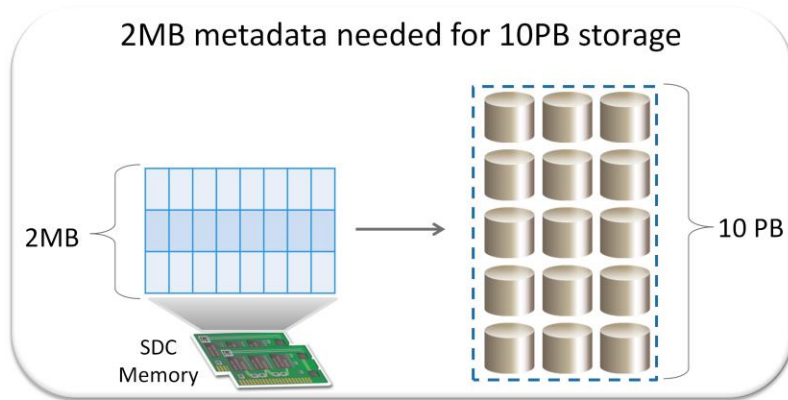
What happens when chunk ownership needs to change for several chunks from one SDS to another?

ScaleIO avoids the traditional approach of first having to contact every client and ensuring that each client knows about the ownership change. This is just not practical in a large system with potentially 100s or 1,000s of clients, some of which may not even be listening.

Instead, it adopts a lazy approach to propagating this metadata change. This means the architecture will enable an SDC to know its metadata is out-of-date and to update it, only on an “as-needed” basis. Similarly, there are efficient mechanisms in place to ensure that each SDS knows exactly when it has ownership of a chunk, and when it does not.

Without such mechanisms, a reliable, large, highly scalable, loosely-coupled cluster cannot be built. And this is why you just cannot assume that a competing product that works well with 10 or 20 nodes will reliably scale out to 1,000s of nodes. Frequently, that will require a complete rearchitecting of their clustering mechanisms.

SDC In-memory Metadata Map



“Lazy” approach to keep in-memory map up-to-date

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

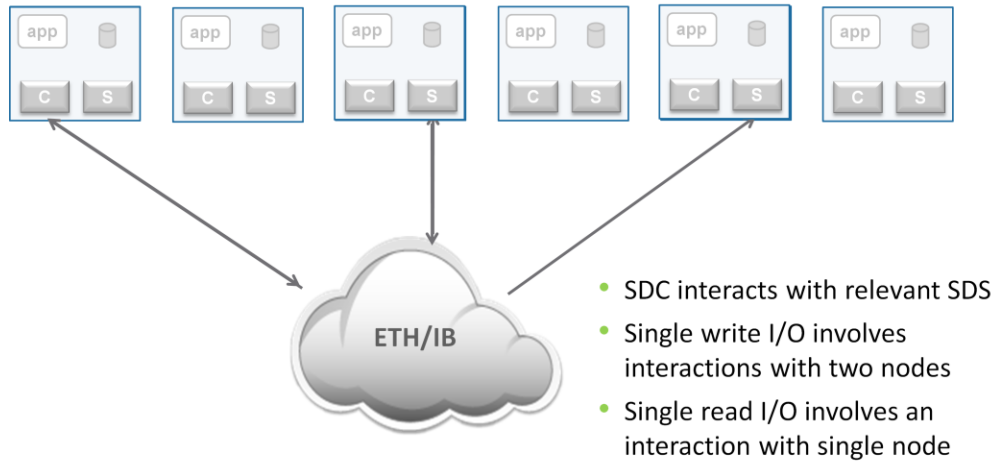
ScaleIO Implementation and Management

24

Each SDC node has an in-memory map that holds information about which SDS owns which chunk. Let's assume that chunk size is 1MB. How many 1MB chunks do we have in 1PB? the answer is 1 billion. ScaleIO can store all needed metadata for 10PB of data in just 2MB of RAM. ScaleIO in fact only has 100s of 1,000s of metadata entries (at most) within this 2MB – not billions. This makes the system highly scalable.

Note that while the MDM maintains metadata on which SDS owns each chunk, the allocation of chunks to specific data locations within an SDS is within the control of that SDS. The SDS maintains its own metadata for this information.

I/O Flow - Theory of Operation



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

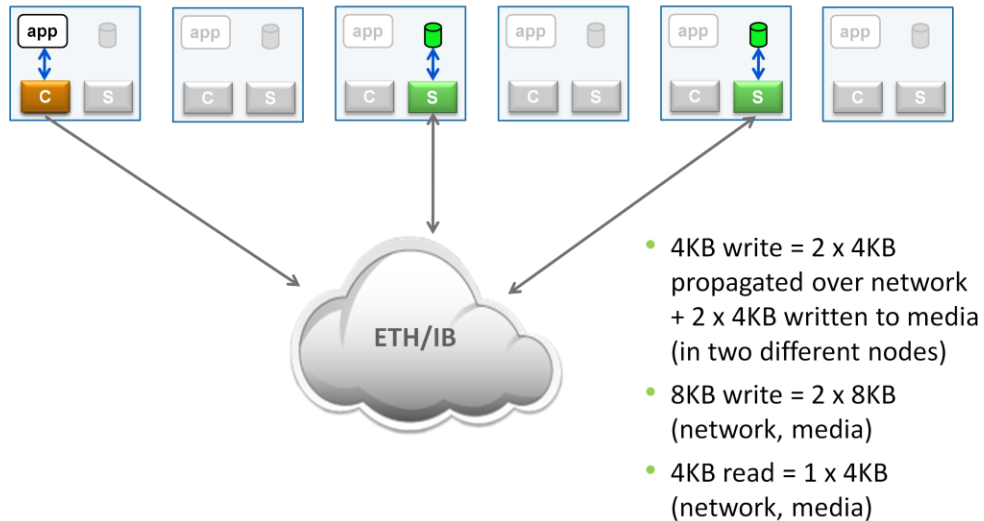
ScaleIO Implementation and Management

25

A read operation typically requires just one interaction between the SDC and one SDS.

A write operation involves two interactions: between the SDC and the primary SDS for that data chunk, and between the primary and secondary SDS to ensure the mirror copy is updated before the write is acknowledged.

I/O Flow - Bandwidth



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

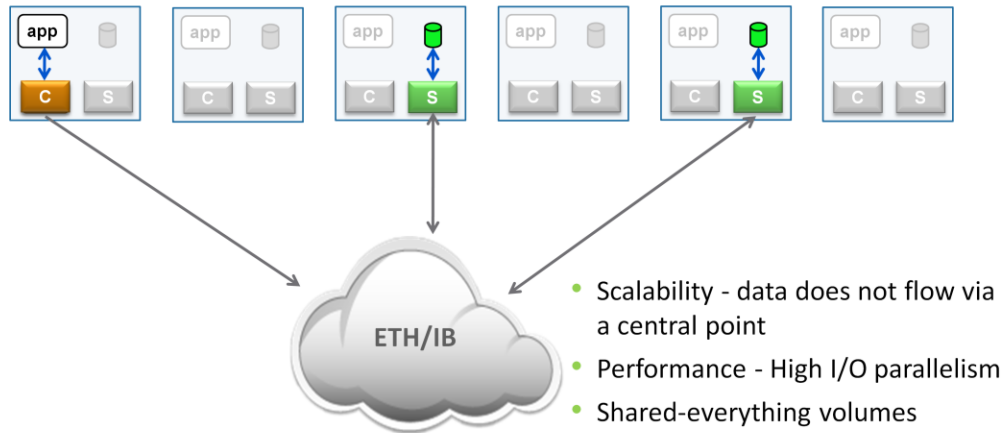
ScaleIO Implementation and Management

26

A read operation results in just the read size being read over the network.

For a write operation, the network overhead is double the write size, since two interactions are required.

I/O Flow - Performance and Scalability



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

27

User data never passes through the MDM. It operates at control rate, not data rate. A single MDM can easily drive 100s or even 1,000s of SDS nodes. The MDM can be viewed as a small, tightly-coupled cluster that manages a large, loosely-coupled SDS cluster.

Performance is ensured for application read and write I/O, since a large number of SDS servers can share the I/O handling. The system behaves like a multi-head storage array with several controllers servicing a single volume in parallel.

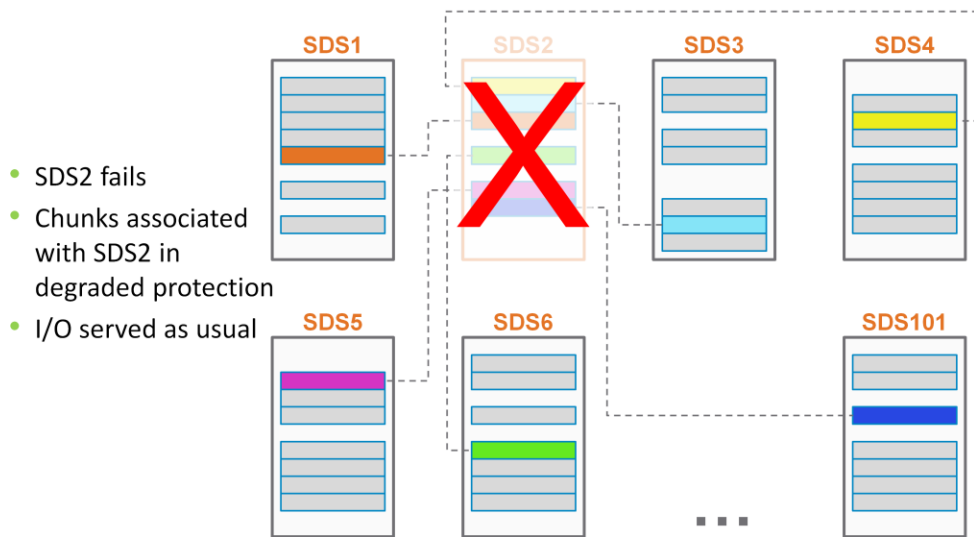
Failure Handling

- Forward rebuild
 - ▶ Once device/node fails, rebuild load is balanced across failure domain devices/nodes → faster and smoother rebuild
- Backward rebuild
 - ▶ Smart and selective transition to “backwards” rebuild (re-silvering) once failed node is back alive



Forward rebuild and backward rebuild are two strategies that ScaleIO uses to ensure ongoing data protection immediately after it detects a node or device failure. Let's examine these strategies in greater detail, next.

Forward Rebuild



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

29

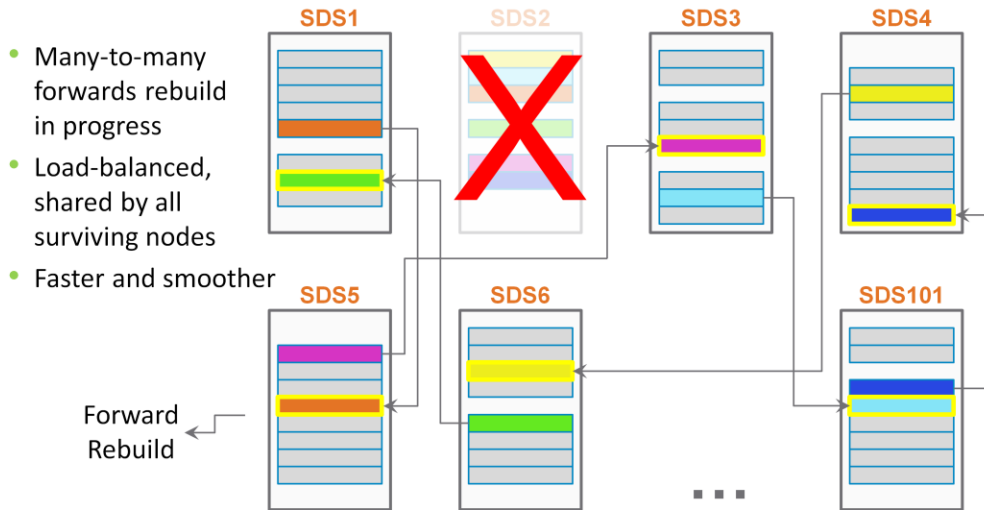
Let's examine carefully what happens when a single node fails.

Consider SDS2, which has chunks that are mirrored across every other node of this 100-node cluster.

When the SDS2 server fails, all I/O requests to ScaleIO will continue to be serviced as usual. However, we have a situation where roughly 1 percent of the data chunks are in what is termed "degraded protection mode", which implies that the chunk does not have a usable mirror.

Upon detecting the failure, ScaleIO will allocate new chunks out of the reserved spare space, to take the place of the failed mirrors. The data gets re-mirrored to the new locations.

Forward Rebuild (cont.)



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

30

Note that remirroring is done in a many-to-many fashion, since the mirrors of SDS2 chunks are scattered across all nodes of the cluster. This will therefore be many times faster when compared to taking the traditional approach of simply restoring the SDS2 node as it used to be – that would be bottlenecked by either the NIC bandwidth of a single node, or the throughput capability of the disk spindles on that single node, and in addition, also be very imbalanced in terms of rebuild load causing degradation in application performance.

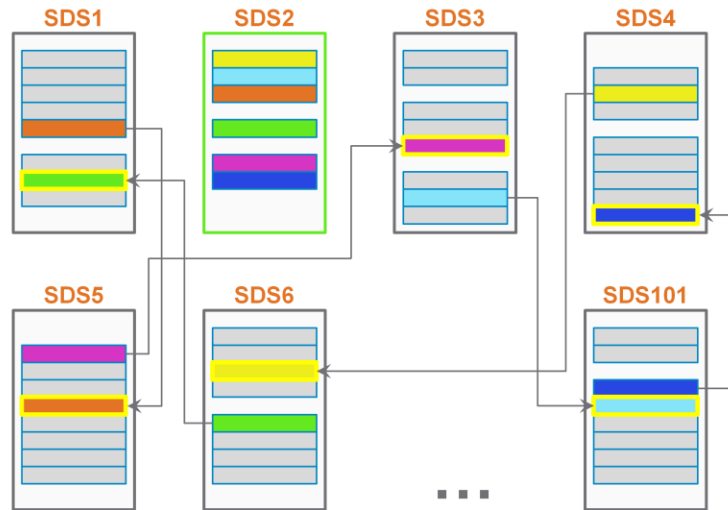
Let's consider a simple example to get a sense of the achievable rebuild time. Assume we have 1TB of data on each of 101 nodes, and a 1 Gbit network in place. When one node fails, we need to re-mirror roughly 1TB of data that was lost. This 1TB of data needs to be propagated between the surviving 100 nodes. That is 10GB of data to be mirrored by each of these nodes. Assuming ~50 MB/sec read speed, it will take about 200 seconds to read the chunks. Let's double that and call it 400 seconds to complete both the reads and writes, which is around seven minutes. This is a pretty realistic example.

Within a cluster "partition", when two nodes go down simultaneously, data could become unavailable. In this example, this unavailability condition will apply only to one percent of one percent - or one 10,000th of the total data (since only the chunks that are mirrored between these two specific nodes are affected). The odds of this happening are quite small, given that the rebuild is many-to-many and quick. The SDCs can continue to access the rest of the available data. Unlike some other competing products, we also have good handling mechanisms in place for when nodes go down momentarily and then comes back. When this happens to two different nodes, the data that becomes momentarily unavailable becomes available again. As you'll see later, ScaleIO has the notion of "Protection Domains" that enable large clusters to tolerate multi-node failures. Nodes that are in separate Protection Domains can fail simultaneously and still not cause any data loss or unavailability.

Backward Rebuild

- SDS2 back alive
- SDS2 chunks contain valid, yet not necessarily up-to-date info
- Forwards rebuild take a while to complete
- ScaleIO decides if SDS2 info still valid

Forward
Rebuild



EMC²

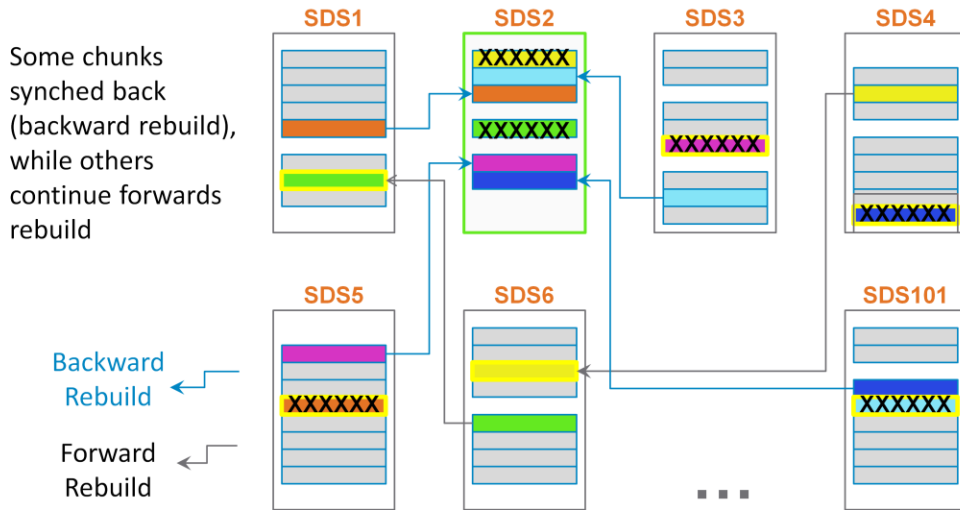
Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

31

What if SDS2 comes back with all its chunks intact after going down for a short while and forward rebuilds are already in progress? Now the situation becomes more complex, because chunks could have been modified *after* the forward rebuild process has started. So some chunks in SDS2 may have out-of-date data at this point. If a chunk has been only slightly modified, it may make sense to simply restore the SDS2 copy from the up-to-date copy elsewhere in the cluster. This is a “backward-rebuild” operation.

Backward Rebuild (cont.)



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

32

ScaleIO is intelligent enough to decide on and do what makes the most sense efficiency-wise - continue with the forward rebuild, or initiate a backward rebuild selectively only for those chunks that have changed. And it can make that decision on a chunk-by-chunk basis.

This design ensures that there is no large penalty efficiency-wise for momentary server failures. A short outage results in only a small penalty, since most of the chunks will either remain unchanged or require a short backwards-rebuild.

Let's look at an example of a forward-rebuild versus backward-rebuild at work. Pretend that a server failed for an hour and was restored.

Meanwhile, 10 percent of the data on that server has been modified due to application writes. Does it make sense to do a forward rebuild or backward rebuild?

It depends on how success is measured. If the goal is to minimize the amount of data being propagated on the network, then the backward rebuild makes the most sense – it would result in only 10 percent of the data being remirrored to SDS2.

However, if the main concern is to reduce the window of vulnerability to a second server outage, the picture changes. You need to remirror 10 percent of the data to a single server. Let's assume there is just one disk spindle per server. With a forward rebuild, you would be propagating ~10 times more data, but it would be to 100 spindles on 100 different servers, so it is going to be 10 times faster.

ScaleIO is designed to target the second goal, that is to “get fully protected as soon as possible”. This tends to favor “many-to-many” copying versus “many-to-one” copying.

Rebalance - Capacity Addition

Nodes or devices may be added dynamically → ScaleIO cluster automatically migrates and rebalances Storage Pool



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

33

Adding a node or device to ScaleIO automatically triggers primary and secondary chunks to be migrated to the newly available devices ensuring balanced utilization of all devices in the corresponding Storage Pool and Protection Domain.

Rebalance - Capacity Removal

Nodes or devices may be dynamically removed → ScaleIO cluster automatically migrates and rebalances Storage Pool



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

34

A similar rebalance mechanism applies to removing a node or device from a ScaleIO system.

Basic Planning and Design Considerations

- Heterogeneous OS environments
 - ▶ Linux: RHEL 6.1+, SLES 11 SP2 +
 - ▶ Windows: 2008 R2, 2012 or 2012 R2
 - ▶ VMWare ESXi 5.5, 6/0
- Network requirement
 - ▶ TCP/IP over Ethernet 1 GbE or 10 GbE (preferred option)
 - ▶ IP over Infiniband
 - ▶ Jumbo frames (9000 MTU) supported
 - ▶▶ Should be consistent across servers, switches and vSwitches
 - ▶ IPv6 not supported
- Check release notes for current HW and SW prerequisites



Shown here are basic planning and design considerations for ScaleIO. Be sure to review all planning and design considerations before attempting to build a solution.

One of the following (for a complete list, see the EMC Support Matrix):

- Linux: CentOS 6.0-7.0, Red Hat 6.0-7.0, SUSE 11 SP2 and SP3, or SUSE 12

Packages required for all components:

- numactl
- libaio

Packages required for MDM components:

- mutt (for Call-Home)
- bash-completion (for scli completion)
- Latest version of Python 2.X
- When installing the MDM component on Linux CentOS 6 or RHEL 6 hosts, set the shared memory parameter in the `/etc/sysctl.conf` file to at least the following value:
kernel.shmmax=209715200. To use this value, type the **sysctl -p** command.

Requirements for running the GUI:

- Java 1.6, or higher
- Screen resolution: 1366 x 768 minimum

- Windows: 2008 R2, 2012, or 2012 R2

Requirements for running the GUI:

- Java 1.7, or higher
- Screen resolution: 1366 x 768 minimum

Packages required for MDM components:

- Install the EMC-provided **PythonModulesInstall.exe** on all MDM nodes. Download the file from the EMC Online Support site (search for ScaleIO Python Installation Modules) on <https://support.emc.com>.

To install SDC on 2008 R2, ensure that Microsoft Security Update KB3033929 is installed.

- Hypervisors:
 - VMware ESXi OS: 5.5 or 6.0, managed by vCenter 5.5 or 6.0 only
 - Hyper-V
 - XenServer 6.1|
 - RedHat KVM

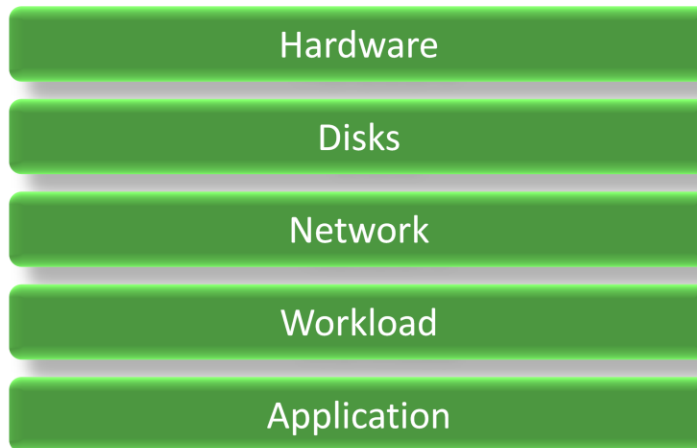
Basic Planning and Design Considerations (cont.)

- Key performance concerns
 - ▶ Memory - light footprint
 - ▶▶ SDC - 20MB RAM
 - ▶▶ SDS - 200 to 400MB RAM (more is using RAM cache)
 - ▶▶ MDM - minimal
 - ▶ CPU - 1 to 2 percent depending on workload
 - ▶ IO and network bandwidth, depending on workload
- Volume capacity < 1PB - thick or thin volumes
- Licenses



These are additional planning and design considerations for ScaleIO.

Key Performance Considerations



Proper solution design is critical to achieve acceptable performance from a ScaleIO cluster. This is a list of the major performance considerations. Hardware capabilities largely determine the performance capability, but it is also important to take into consideration per-node limitations when they are a factor. The application read/write profile and workload determines the performance requirement.

Let's examine each of these factors in greater detail.

ScaleIO and Media

- Supports any type of media
 - ▶ HDD, SSD, PCIe
 - ▶ Internal and external
- Recommended: Internal RAID controller
 - ▶ Benefit: Exploit hardware cache for performance
 - ▶ For reads and writes (if battery backed)
- Can leverage unused partitions on devices
- Can use free space on existing file system
 - ▶ Create and use special file on file system as ScaleIO device
 - ▶ File system overhead not serious concern, if using this approach
- Maximum of 64 devices per SDS



ScaleIO is designed for maximum flexibility in terms of choices for pooled devices.

Any type of random-access device on a host can be included in a ScaleIO pool. This includes everything from PCIe Flash cards to SSDs to local HDDs of all types. SAN attached devices are also supported, for example, when transitioning out of an existing traditional SAN.

In addition, partitions of existing drives can be used. It is also possible to leverage the spare capacity in an existing file system and use it as a ScaleIO device with minimal file system overhead.

Sizing for Capacity and Sparing

- Assume
 - ▶ Servers: n
 - ▶ Capacity per server: c TB
 - ▶ Number of server failures to tolerate: k
- Available capacity
 - ▶ Spare: Remove server capacity: $(n - k) * c$
 - ▶ RAID: Two copies

$$S = \frac{(n - k) * c}{2}$$

- ▶ **Example: 11 servers, 3TB each, tolerate one failure**

$$S = \frac{(11 - 1) * 3}{2} = 15\text{TB}$$



Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

40

The example shows a simple calculation to estimate the available protected capacity in a ScaleIO cluster.

In an 11-node cluster, assuming the system needs to protect against just one failed node, the protected capacity available to ScaleIO is 15 TB (out of the aggregate raw capacity of 33 TB).

It is possible to protect against multiple node failures (with the constraint that the next node fails only after the system has recovered from the prior failure with a completed rebuild action). In this example, let's assume the requirement is to protect against three failed nodes. Then the available protected capacity drops to: $(11 - 3) * 3 / 2 = 12$ TB.

The required degree of sparing is a design consideration, and it can be configured on a ScaleIO cluster using the "%" sparing parameter.

ScaleIO Performance: General Caveats

- All numbers presented here are qualitative
- Real-life performance depends on many factors:
 - ▶ Specific hardware configuration
 - ▶ Specific workload
- Engage a ScaleIO expert or perform actual testing, to validate results of any qualitative performance analysis



We will now briefly cover design considerations relevant to ScaleIO performance.

In this presentation, note that all numbers presented are qualitative and should be treated as such! In real life situations they depend on hardware type, actual workload and more. Nothing replaces real performance analysis by a ScaleIO expert or actual testing!

Performance - Limiting Bus Speeds

- Internal network
 - ▶ PCIe: 16GB/sec
 - ▶ SATA/SAS: 600MB/sec
- External network
 - ▶ 1 GbE
 - ▶ 10 GbE



Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

42

Regardless of the drive capabilities, it is important to note the bus limitations within each server. For example, a SATA or SAS bus will bottleneck at 600MB/sec aggregate throughput. Adding drives to a single bus will not deliver any additional bandwidth.

The same applies to the overall network capability. A given 1 GbE or 10 GbE network can only deliver that level of aggregate bandwidth. Once the network saturates, adding servers or devices will not deliver more throughput.

Performance - Device Capabilities

Device Type	IOPS Read	IOPS Write	Bandwidth Read MB/Sec	Bandwidth Write MB/Sec
HDD	100	100	100	100
HDD/Cache	100	350	100	100
SSD	50 K-70 K	12 K	400	250
PCIe-Flash	100 K	60 K	700	700

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

43

The type of storage device will dictate the per-device performance. Note that HDD will benefit mainly from improved write IOPS when a host-based RAID controller is used. This assumes write-back cache is used with battery backup.

When aggregating throughput on a single server from many devices, make sure to take into consideration the bus limits.

Performance - SDC and SDS Capability

	IOPS	Bandwidth
SDC	100 K	2 to 3 GB/s
SDS	100 K	2 GB/s

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

44

Beyond media capabilities, there are limits to what a single SDC and a single SDS can typically deliver, in terms of IOPS and bandwidth. These limits will constrain the aggregate IOPS and bandwidth that can be delivered in a small ScaleIO clusters with low node counts. With ScaleIO, performance actually benefits from scaling out by adding nodes.

Aggregate Performance - Design Checklist

- Assess workload
 - ▶ Reads: IOPS and BW (same as application read rate)
 - ▶ Writes: IOPS and BW (*double* application write rate)
- Estimate per-SDS capability: Sum IOPS/BW of devices
 - ▶ Account for internal bus limits
 - Protection Domains
 - Storage Pools
- Scale out by number of SDS servers
 - ▶ Account for Protection Domains, Storage Pools
- Check network bandwidth
 - ▶ Plan for rebuild/rebalance traffic



Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

45

This is a checklist for performance-related design calculations.

First, assess the workload in terms of needed IOPS and bandwidth. This should be done for both reads and writes.

Second, estimate what each SDS can deliver per Protection Domain and per Storage Pool.

Third, scale out the per-server numbers by multiplying by the node count. This calculation may need to be more elaborate with servers organized into multiple Protection Domains, and devices organized into Storage Pools within Protection Domains.

Fourth, use the estimated aggregate bandwidth to appropriately size the LAN infrastructure. This can help determine if a 1 GbE infrastructure that is already in place is adequate, for example. In this computation, the designer may want to build in some extra network capacity for rebuild and rebalance load, especially in dynamic environments with high node counts.

Assessing Performance - Example 1

Small Block writes with HDD: IOPS-intensive

- Block Size 8KB
- Average write workload: 100 IOPS
- Per server: 100 IOPS
- Total eight servers: 800 IOPS
- Throughput: $800 \times 8\text{KB} \times 2$:
 - ▶ $6.4\text{MB/Sec} \times 2 = 12.8\text{MB/Sec}$
 - ▶ Network: 1 GbE



Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

46

In our first example of a simple environment assessment, we considered a write IOPS-intensive environment with a small Block size (8KB). The computation shows that with eight servers doing 100 IOPS each, ScaleIO will consume about 10 percent of the available network bandwidth. So, if a GbE network is already in place, it can probably be leveraged.

Assessing Performance - Example 2

Large Block reads with SSD: Throughput-intensive

- Reads per server: 400MB/Sec
- Total eight servers: 3.2GB/Sec
- Single 10 GbE segment can deliver ~1GB/sec
 - ▶ Do we need multiple networks? Yes.
 - ▶ What about the 2GB/sec limit of a single SDS? Does not apply (server generates 400MB/sec in this case)

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

47

Our second example is read bandwidth-intensive with SSDs and shows a requirement of 3.2GB/sec.

A single 10 GbE network can deliver 1.25GB/sec at peak, which is less than the required 3.2GB/sec.

Licensing

- non-production environments
 - ▶ Unlimited time, capacity
 - ▶ Unsupported
- Licensed options for production
 - **Basic:**
Enables all basic functionality
 - **Enterprise:**
Enables advanced features
- Getting a license
 - Open a support ticket
 - `set_license` command

Feature/Functionality	Basic	Enterprise
High Availability	Yes	Yes
Self-Healing	Yes	Yes
Scale-out architecture	Yes	Yes
Support for thousands of nodes	Yes	Yes
Elasticity	Yes	Yes
Hyper Convergence	Yes	Yes
Automated performance tuning	Yes	Yes
Any-Drive support: SSD, PCIe, HDD	Yes	Yes
Automatic Cluster Growth	Yes	Yes
Asymmetric nodes support	Yes	Yes
Advanced Management	Yes	Yes
REST API	Yes	Yes
OpenStack Support	Yes	Yes
vSphere Plug-in	Yes	Yes
QoS		Yes
Obfuscation		Yes
Snapshots		Yes
Auto tiering (Flash caching)		Yes
RAM caching		Yes
Fault Sets		Yes
Thin provisioning		Yes

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

48

ScaleIO installations are enabled to be fully functional, for non-production environments.

Using ScaleIO in a production environment requires a license. The license is installed on the MDM cluster, using the `set_license` command.

To obtain a license for production use, and to receive technical support, open a service ticket with EMC Support at <https://support.emc.com>.

ScaleIO supports the following license options:

- **Basic:** Enables all basic functionality
- **Enterprise:** Enables advanced features

For more information about the features enabled per license type, see the ScaleIO Licensing Options Technical Note, available for download from EMC Online Support. The license is installed on the MDM cluster, using the `set_license` command. ScaleIO installations are enabled to be fully functional, at the enterprise level, for the first thirty days after installation. After this time elapses, the system will continue to operate, however no configuration changes will be allowed until a permanent license is installed.

Solution Design - Best Practices

- Size system properly for adequate capacity
 - ▶ Consider sparing: how many failures will be tolerated within single Protection Domain?
 - ▶ Consider space required for mesh mirroring
 - ▶ Thin versus thick: level of over-subscription
- SDS nodes per Protection Domain
 - ▶ Try not to exceed 128 nodes per Protection Domain
 - ▶ Multiple Protection Domains => higher MTBF, more resilience
- Storage Pools
 - ▶ Avoid mixing drive types within single Storage Pool
 - ▶▶ This would reduce performance of the pool to slowest device type
 - ▶ Match drive capacities across drives in pool and SDS servers whenever possible



Several design best practices for a ScaleIO solution are listed here.

Positioning and Target Use Cases

- VSI
 - ▶ Enterprise Private Cloud
 - ▶ Service Providers, Telco
- Test and development
 - ▶ In-house alternative to offloading test and dev to cloud
- Databases
 - ▶ SAP HANA
 - ▶ MS SQL
 - ▶ Oracle RAC



Shown here are the major use cases for ScaleIO.

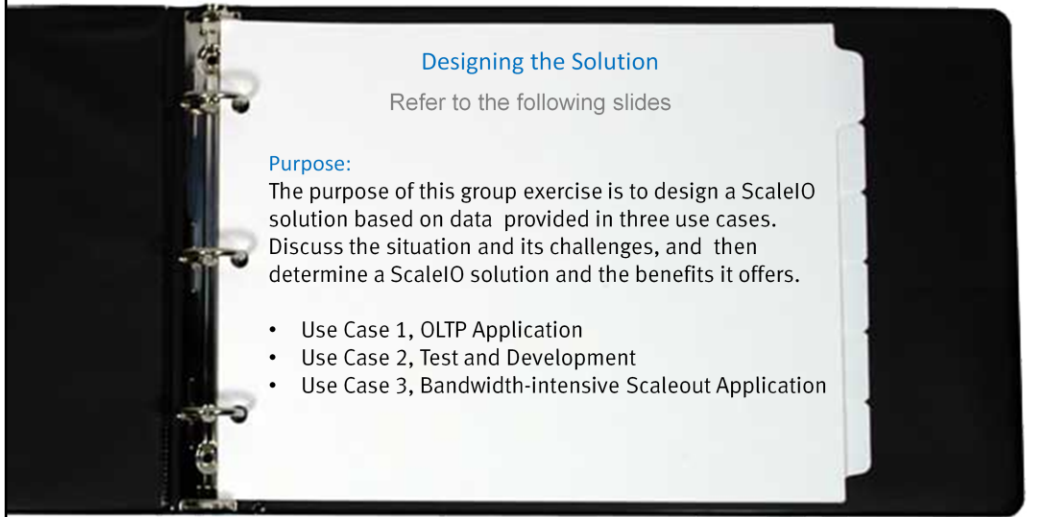
ScaleIO Enablers for Multi-tenancy/Service Providers

- Protection Domains enable physical segregation
- LUN mapping enables local segregation
- Storage Pools/Protection Domains provide storage tiering
 - ▶ Guarantee service level for each tier
- Ability to move drives across Storage Pools and SDS nodes across Protection Domains on demand
 - ▶ Elasticity to handle dynamically shifting workload across tenants
- IOPS and bandwidth limit feature
 - ▶ Per volume, for given SDC accessing that volume
 - ▶ Limit service to tenant, based on tier



Here are the key ScaleIO features enabling multi-tenancy.

Group Exercise



Designing the Solution

Refer to the following slides

Purpose:
The purpose of this group exercise is to design a ScaleIO solution based on data provided in three use cases. Discuss the situation and its challenges, and then determine a ScaleIO solution and the benefits it offers.

- Use Case 1, OLTP Application
- Use Case 2, Test and Development
- Use Case 3, Bandwidth-intensive Scaleout Application

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved. ScaleIO Implementation and Management 52

Please proceed to the classroom group exercise use cases, “Designing the Solution”. This is not included in the Lab Guide. The class will be divided into groups and assigned a case. The objective is to discuss the current situation of the case and its challenges, and then determine a ScaleIO solution and the benefits it provides.

1. Use Case 1 - OLTP Application
2. Use Case 2 - Test and Development
3. Use Case 3 - Bandwidth-intensive Scaleout Application

Use Case 1 - OLTP Application

Current Situation

- IAAS Hosting Corp looking for new storage platform to support three existing customers and future new customers
- Three OLTP clients have the following IO profiles (typical Block size=4 Kbytes for all):
 - Client-A has 2TB database that requires 70K IOPs and has an 80/20 RW ratio
 - Client-B has 1TB database that requires 30K IOPs and has an 60/40 RW ratio
 - Client-C has 1TB database that requires 40K IOPs and has an 70/30 RW ratio
- Current database update process requires at least 20TB of common shared storage
- No immediate concerns about data isolation by customer
- IAAS needs to plan for 100% growth in IOPs and required capacity
- IAAS has dedicated 200 new x86 servers (24x 600GB 15K rpm drives each) to service all OLTP

Challenges

- The OLTP applications are both compute-intensive and RAM-intensive: require minimal CPU and RAM overhead on host for any proposed solution
- Ability to scale given anticipated rapid growth
- Escalating CAPEX and OPEX costs with existing storage frames



Use Case 1 - Solution and Benefits

Solution

- Deploy EMC ScaleIO with fully-converged topology with all 200 OLTP servers as SDC+SDS
- Configure two Protection Domains - 100 servers each (follow ScaleIO best practice)
- Consider dual 10-Gbit NICs for ScaleIO use on all 100 servers (to accommodate 100% growth)
- Consider host-based RAID with write-back cache for best performance
- Configure single Storage Pool per Protection Domain
- Manually balance clients across Protection Domains to ensure performance

Benefits

- Fully leverages existing compute and internal storage, reducing CAPEX costs
- Highly scalable solution ideal for Service Provider with growing customer base
- Available features provide data isolation by customer, if it becomes a requirement



Use Case 2 - Test and Development

Current Situation

- Mid-size bank with in-house software development of various applications
- All applications have been migrated over time to x86 platforms
- QA team uses the same test data sets as the development team
- Persistent test coverage issues with QA unable to keep up-to-date with changes in data sets
- 100s of x86 servers with existing production storage on legacy NAS

Challenges

- Dev team and QA team must maintain separate copies of each test data set
- QA updates its copy overnight from dev using file copies over NFS
- Data sets have been growing over time – it takes too long to perform file copies
- QA and dev both want more control over storage provisioning and management, but have no meaningful SAN skills



Use Case 2 - Solution and Benefits

Solution	
Benefits	



Use Case 3 - Bandwidth-intensive Scaleout Application

Current Situation

- Analytical Wizards has just won a contract from the State
- Single IT facility will support the capture and processing of real-time data
- Facility has 1/2 Gb Fibre Channel deployed; switch port count is at capacity
- Current server/storage infrastructure is 90% capacity with little room for growth
- Analysis application requires new 3-node Microsoft Hyper-V cluster
- Customer is open to a green field solution

Challenges

- Analysis: Application running on Hyper-V cluster - requires shared storage
- Data Collection: currently 600 data feeds writing 5Mbps each
- Captured data kept for seven days after which external application removes it
- Initial purchase must accommodate 50% capacity growth
- Must be able to quickly add compute and storage per processing demands
- Ensure captured data remains online in a highly available architecture
- IT staff already working more than 40 hrs. with no new staff to support contract



Use Case 3 - Solution and Benefits

Solution	
Benefits	



Module 2: ScaleIO 1.30 Installation

Upon completion of this module, you should be able to:

- Use the ScaleIO Installation Manager (IM) to bring up a ScaleIO cluster with physical hosts (Linux, Windows)
- Deploy and use the ScaleIO vSphere Web Client plug-in to install ScaleIO in a VMware environment



This module focuses on initial installation of a ScaleIO cluster and the subsequent dynamic growth of that cluster by adding nodes non-disruptively.

Installation Manager Overview

ScaleIO Installer

Home Packages **Install** Monitor Maintain

Provide Installation Topology

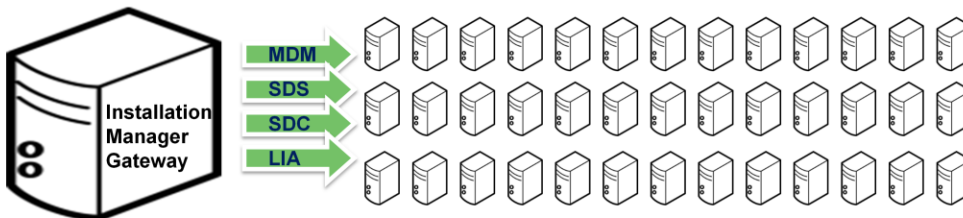
Upload an installation .csv file that describes the desired topology of the new ScaleIO system. Examples for the .csv file: [Minimal](#) & [Complete](#)

Browse ... New Installation Upload Installation CSV

OR

Perform a simplified installation of a **new** ScaleIO system using common defaults:

Installation wizard



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

60

Physical Hosts - Installation Manager Overview

Supported operating systems	One of the following: • Windows 7, 2008 R2, or 2012 R2 • Linux RHEL 6.x or 7.0, SUSE 11 SP2 and SP3, or SUSE 12 Every server requires 2 cores and a minimum of 2 GB RAM.
Supported web browsers	The web client is supported on the following browsers: • Internet Explorer 10, or higher • Firefox, version 37, or higher • Chrome, version 41, or higher
Java requirements	• Linux—1.6 or higher, 64-bit • Windows—1.7 or higher, 32 or 64-bit (for better performance)
Other	• For a Windows Installation Manager (IM), the WMI service must be enabled on the IM server and on all Windows ScaleIO nodes. • The Gateway server must have connectivity to all the nodes that are being installed. If you are using separate networks for management and data, the server must be able to communicate with both networks.



The ScaleIO IM can be installed on either a Linux or Windows platform.

Once installed, it can be accessed from either a web interface or the command line.

For installing a cluster, the IM requires a user-supplied CSV file, which is easily viewed and modified.

Input CSV File - Minimal and Complete Formats

CSV templates available from IM GUI

ScaleIO Installer Home Packages **Install** Monitor Maintain **EMC ScaleIO**

Upload Installation CSV
Upload an installation .csv file that describes the desired topology of the new ScaleIO system. Examples for the .csv file: **Minimal** **Complete**

Browse ... New installation Upload installation CSV

ScaleIO_Minimal_Config.csv

IPs	Password	Operating Is	MDM/TL	Is SDS	SDS Devic	Is SDC
10.76.60.1	Password: linux	Primary	Yes	/dev/sdb	Yes	
10.76.60.2	Password: linux	Secondary	Yes	/dev/sdb	Yes	
10.76.60.3	Password: windows	TB	Yes	/dev/sdb	Yes	

ScaleIO_Complete_Config.csv

Domain	Username	MDM Mgmt IP	MDM IPs	SDS Name	SDS All IPs	SDS-SDS Only IPs	SDS-SDC Only IPs	Protection Domain	Fault Set	SDS Pool List	Optimize IOPS
root		10.76.1.1	10.76.60.195		10.76.60.195			domain1	fs1	pool1	Yes
localhost	administrator		10.76.60.17								Yes
root		10.76.60.33	SDS1	10.76.60.33	10.76.2.2,10.76.4.4	10.76.3.3		domain1	fs1	pool1	Yes

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved. ScaleIO Implementation and Management 62

Shown here are the “minimal” and “complete” CSV templates downloaded from the IM.

The “minimal” CSV file contains fields for only the mandatory information required to set up a simple three-node cluster. This includes none of the optional parameters.

The “complete” CSV file, too wide to be displayed in full here, provides the opportunity to customize the cluster by specifying any of the optional parameters.

CSV File - Installation Parameters for Each Host

MANDATORY parameters	OPTIONAL parameters
• IP address • Password of privileged user • Operating System • Is MDM/TB (yes/no) • Is SDS (yes/no) • SDS Device List • Is SDC (yes/no)	• SDS Name • SDS All IPs • SDS-SDS Only IPs • SDS-SDC Only IPs • Domain • Fault Set • SDS Device List • SDS Pool List • Optimize IOPS (yes/no)

- Optional parameters set automatically to default values, if not provided in input CSV file
- IM checks CSV file for errors



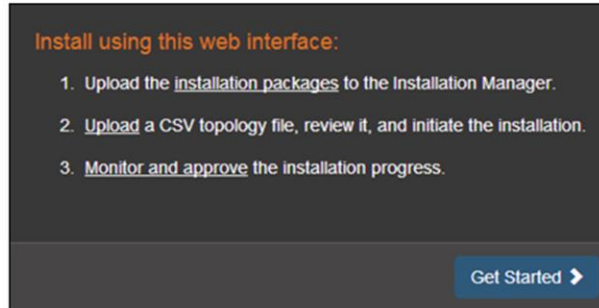
The installer can choose from one of two different formats when creating an input CSV file: a “minimal” or a “complete” CSV file format. The “minimal” CSV file format contains fields to input only the mandatory parameters, as shown here. The “complete” CSV includes all the additional optional parameters. Both formats can be downloaded directly from the IM.

During the upload of the CSV file, the IM checks for completeness with regards to the mandatory parameters, and parses all parameters for acceptable values.

If the CSV file contains any errors, the IM will refuse to complete the upload. Typically, the error message will attempt to point the installer to the missing or erroneous field(s).

Installation Steps for IM Web Interface

- 1 Log in to IM server
- 2 Upload ScaleIO RPM/MSI packages to IM server
- 3 Upload CSV file
- 4 Perform installation



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

64

The major steps for installing a ScaleIO cluster using the web interface are shown here.

Note that the installer is required to configure a separate password to access the Installation Manager itself. Before any installation can be performed using this interface, authentication with these credentials is required.

Installation Steps for IM CLI

- 1 Download and launch CLI
- 2 Log in to IM server
- 3 Upload ScaleIO RPM/MSI packages to IM server
- 4 Run **im install** command with the CSV file as input



Shown here are the four major steps to accessing and using the existing Installation Manager via the CLI.

IM Capabilities - Web UI vs. CLI

Operation	Web Interface	IM CLI
Install from CSV	✓	✓
Upgrade via LIA	✓	✓
Get Info via LIA	✓	✓
Uninstall via LIA	✓	✓
Uninstall from CSV		✓
Upgrade from CSV		✓
Get Info from CSV		✓
Configure Call Home	✓ Only during installation	✓ Only after installation

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

66

The IM CLI provides more complete functionality than the web UI. Specifically, uninstall and upgrade operations can only be performed using the IM command line.

Installation Wizard

Credentials Configuration

MDM Password: [password field] [password field] [?]

LIA Password: [password field] [password field] [?]

License: ☒ I accept the terms of the End User License Agreement (mandatory). [View EULA](#)

Topology

Click on a table cell to edit (except for the role which is fixed).
Press <ENTER> to apply changes.

Nodes

Role	IP	OS	Password
MDM Primary / SDS / SDC	10.0.0.1	Linux	
MDM Secondary / SDS / SDC	10.0.0.2	Linux	
Tie Breaker / SDS / SDC	10.0.0.3	Linux	

Viewing 1 - 3 of 3

+ Clone host ✕ Delete host ☐ Show passwords while editing

Start Installation > - OR - ✕ Cancel

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

67

This abbreviated mode enables you to get a ScaleIO system up and running in the simplest manner, with preset node configuration, where all management and data communication are on the same network. This mode is perfect for a single Protection Domain, fully-converged system. No CSV file is required.

Lightweight Agent (LIA)

- Enables upgrades, uninstalls and **get-info** operations
- Configuration file provides fine tuning to control supported operations

▶ Example (`/opt/emc/scaleio/lia/cfg/conf.txt`):

```
lia_enable_install=1
lia_enable_uninstall=1
lia_enable_configure_call_home=1
lia_enable_fetch_logs=1
```

Note: All operations are enabled by default (=1)

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

68

Version 1.30 introduces the ScaleIO Lightweight Agent (LIA) that can be optionally installed on every cluster node. This is strongly recommended, since the use of LIA assists with system security in two ways. First, the LIA uses its own credentials set by the person performing the ScaleIO installation. This implies that LIA-managed actions do not require the root or administrator password of each cluster node. And second, the LIA will only install or upgrade packages that have been signed by ScaleIO.

SDS - Server Disks Types

- Linux
 - ▶ Raw space
 - ▶▶ /dev/sde
 - ▶ Partition
 - ▶▶ /dev/sde1
- Windows
 - ▶ Drive letter
 - ▶▶ E:
 - ▶ Physical drive
 - ▶▶ \\.\PHYSICALDRIVE3
 - ▶ File space
 - ▶▶ C:\sio (file is created using the **create_file_storage.exe** command)



Shown here are the different types of disks that can be presented to a ScaleIO pool.

Extending a Cluster

- Nodes can be added to an existing cluster in any of these ways:
 - ▶ Using IM with a modified CSV file
 - ▶ Using ScaleIO GUI
 - ▶ Manually, using SCLI commands to MDM (same as v1.21)



ScaleIO v1.30 offers several ways for dynamic growth of a cluster by adding nodes.

For adding several nodes in one automated step, the use of a CSV file and IM is recommended.

For small simple modifications, for example, a single SDS, use either the ScaleIO dashboard, or make a manual addition using the SCLI commands.

Lab Exercise

Lab 2a: Install and Expand a ScaleIO Cluster on Linux/Windows Hosts

Refer to the *Student Lab Guide*

Purpose:

The purpose of this lab is to use the ScaleIO v1.30 IM graphical utility to install and expand a ScaleIO cluster on physical platforms running a mix of Linux and Windows.

EMC²

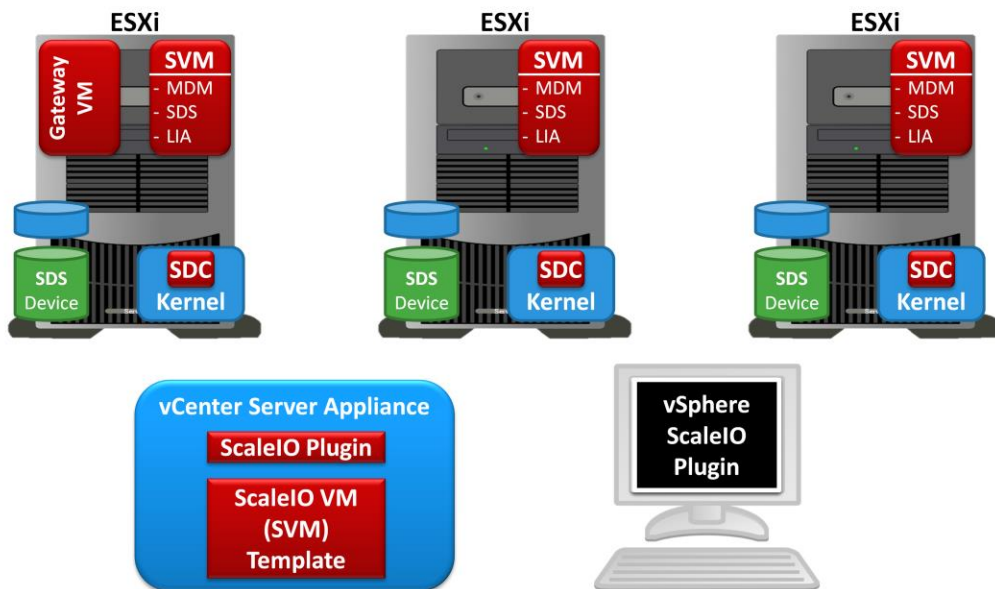
Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

71

Please proceed to Lab Exercise 2a, Install and Expand a ScaleIO Cluster on Linux/Windows Hosts.

VMware Integration Overview



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

72

1. A PowerCLI script is used to:
 - install (or uninstall) the ScaleIO plugin onto a vCenter Server
 - Install the ScaleIO VM (SVM) template into a vcenter Datacenter / Datastore
2. Then, the ScaleIO Plugin is run from within the vSphere Web Client
 - a) First the SDC must be installed on the ESXi servers. It runs in the ESX kernel and requires a reboot to become functional
 - b) Then, via a deployment wizard:
 - ▶ SVM's are installed onto every participating ScaleIO member.
 - ▶ Three are configured for MDM (pri, sec, tb)
 - ▶ Protection Domains and Storage Pools can be configured
 - ▶ SDC's (previously configured) are added
 - ▶ SDS's are chosen, Devices can also be added at this time:
 - ▶▶ RAW devices which are passed through to the SVM via RDM
 - ▶▶ Alternately VMDKs can be created and used (in specific environments)
 - ▶ LIA is installed on all SVMs
 - ▶ A single, separate gateway VM is configured to enable upgrade functionality

Demo: VMware installation



Requirements for ScaleIO with VMware

- VMware version 5.5 or 6.0
- Three or more ESXi servers
- Aggregate storage for SDS use - 300GB or more
 - ▶ Note also: 100 GB minimum per SDS disk
- vSphere Web Server to manage environment
 - ▶ vSphere Web Client that requires:
 - ▶▶ Supported browser - Internet Explorer 7, 8 or Firefox 3.6
 - ▶▶ Adobe Flash Player 10.1.0 or later
- VMware PowerCLI installed on client host
- ESXi datastore to host SVM template



ScaleIO with VMware - Key Deployment Considerations

- Single network for all traffic, or separate networks for management traffic and data traffic?
- Single data network or redundant (two) data networks?
- Data network(s) already in place or need to be created?
 - ▶ Plug-in can create required vSwitches and iSCSI HBAs automatically, if needed
- How will physical storage be presented to SVMs?
 - ▶ VMDK works with all physical devices
 - ▶ RDM recommended - only for devices behind RAID controller
- Should ScaleIO be optimized for Flash devices, if any?



Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

75

While the installation process itself is quite simple, it's critical to know a few details about the target environment in advance.

Specifically, the installer should know whether separate networks are available so that management traffic can be segregated from data traffic for security, efficiency and performance. The plug-in can support automated installation for any of the following scenarios:

- Single network to carry all management and data traffic
- Two networks with segregation of management traffic and data traffic
- Three networks; one network for management traffic, and two redundant networks dedicated to data traffic

If the VMware environment has already been configured with iSCSI provisioning in place, the existing data network(s) can be leveraged and used as-is.

If NICs are available on each ESX server cabled into the infrastructure, but otherwise un-configured, the install procedure can optionally create vSwitches as needed to support one or two separate data networks.

For storage presentation to the SVM, the installer must pick either RDM passthrough or VMDK as the presentation mechanism. Note that RDM passthrough, while recommended, cannot be supported with certain types of local storage devices on an ESXi host. Specifically, it is allowed only for devices behind a RAID controller. When this occurs, the VMDK option can be used instead for device presentation. With the VMDK option, the install procedure automatically builds the required datastores and VMDK files on the selected devices.

The install procedure also offers a means to signal to ScaleIO, if there are Flash devices on a particular SDS so that it can optimize for Flash.

Lab Exercise

Lab 2b: Install a ScaleIO Cluster on VMware Hosts

Refer to the *Student Lab Guide*

Purpose:

The purpose of this lab is to use the ScaleIO v1.30 vCenter Web Client plug-in to deploy a ScaleIO cluster on ESXi servers.



Please proceed to Lab Exercise 2b, Install a ScaleIO Cluster on VMware Hosts.

Module 3: ScaleIO 1.30 Configuration and Management

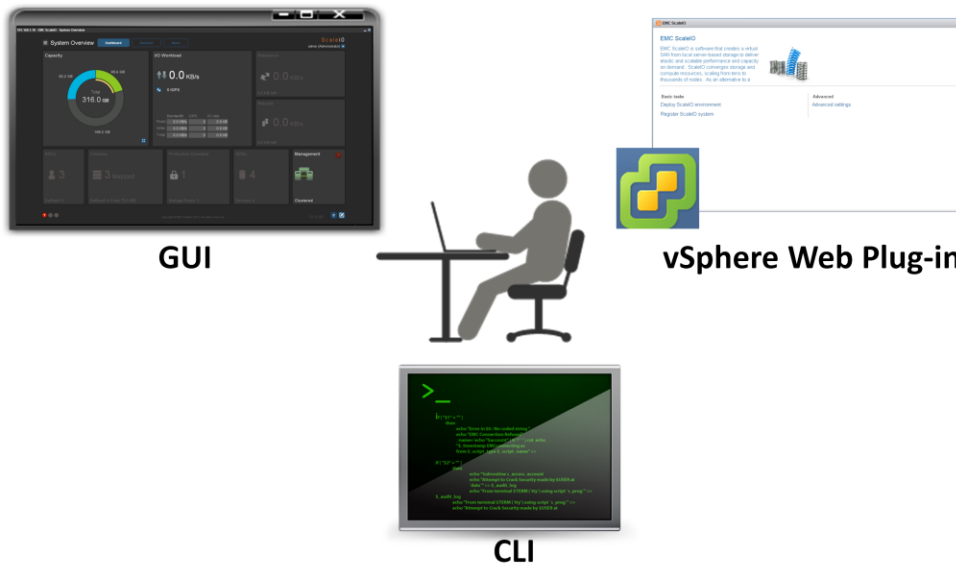
Upon completion of this module, you should be able to:

- Perform basic configuration and management tasks in a ScaleIO environment
- Use any appropriate ScaleIO management tool (GUI, CLI, vSphere web plug-in) to verify and modify a ScaleIO cluster configuration



This module focuses on day-to-day configuration and management tasks in a production ScaleIO cluster. ScaleIO offers multiple user interfaces to accomplish these tasks.

ScaleIO Management Tools



Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

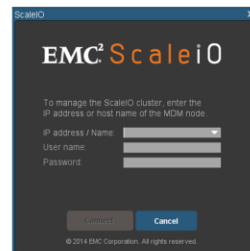
78

EMC²

ScaleIO ships with a command-line interface that can be used to perform any available configuration and management operation. In addition, Version 1.30 introduces a revamped GUI that extends beyond basic monitoring functionality, enabling some of the simpler administrative tasks to be performed. For VMware environments, Version 1.30 also introduces a plug-in for the vSphere web client, allowing VMware administrators to perform basic ScaleIO management directly from the Web Client.

ScaleIO GUI

- Installed separately after cluster deployment
- Can be accessed via
 - ▶ Windows → Start > All Programs > ScaleIO > EMC ScaleIO GUI
 - ▶ Linux → `/opt/emc/scaleio/gui/run.sh`
 - ▶ Mac OS → Copy over and run JAR file
- Designed to work responsively with large-scale ScaleIO clusters
- One instance of GUI per ScaleIO cluster



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

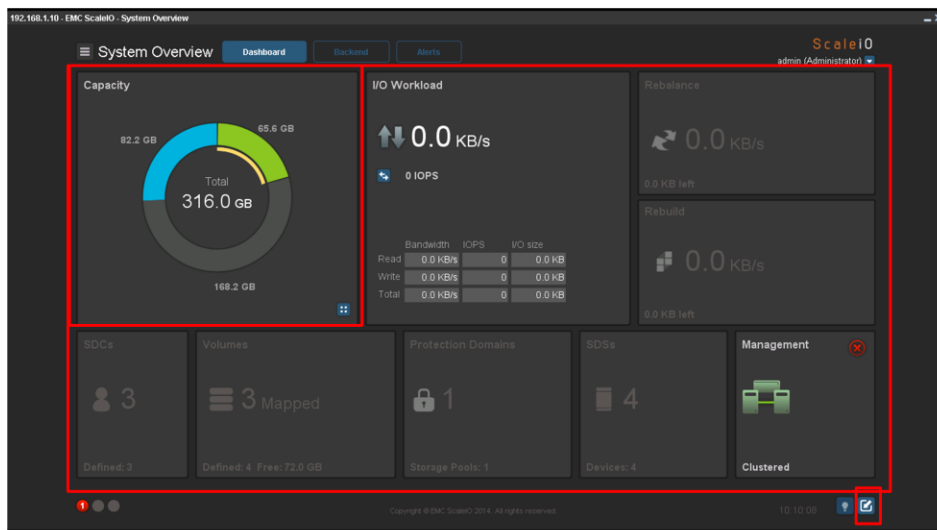
79

ScaleIO 1.30 offers an “active” GUI that ships with the software and needs to be installed separately after the cluster is deployed. It is a Java-based utility requiring Java version 1.7 or higher.

The GUI enables you to perform many standard maintenance activities, as well as monitor the storage system health and performance. You can use the GUI to view the entire system and then drill-down to different elements.

The GUI is designed with large-scale clusters in mind and works effectively over high-latency networks. It communicates with the MDM to accomplish configuration changes and obtain data for display. It can work in conjunction with other interfaces, such as the CLI, vSphere Web Client and REST API. Note, however, that only one instance of the GUI can be active per ScaleIO cluster.

ScaleIO GUI - Dashboard View



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

80

The dashboard displays the overall system status. Each tile displays a certain aspect of the storage system. Various controls let you customize the information displayed on the dashboard.

In the Capacity tile, note that it is possible to get the legend to show what each color represents for the way that storage is used. The I/O Workload tile allows you to display the IOPs or bandwidth, depending on your preference.

In general, note that only the tiles with “interesting” data are highlighted, while the other panels are dimmed. For example, here notice that there is no rebalance and rebuild activity currently in the system and the entire tile is dimmed. As another example, the “Management” tile would be dimmed, if everything were normal and the MDM cluster were running in clustered mode. If it is set to “single” mode, then it would be highlighted, since the ScaleIO GUI wishes to alert the user to that fact.

ScaleIO GUI - Dashboard Mini View



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

81

You can use the “mini-widget” button on the lower right of the screen to get a mini-view of some particular tile, if you need to conserve on the display area required during routine monitoring. For example, you can choose to display only the Capacity tile.

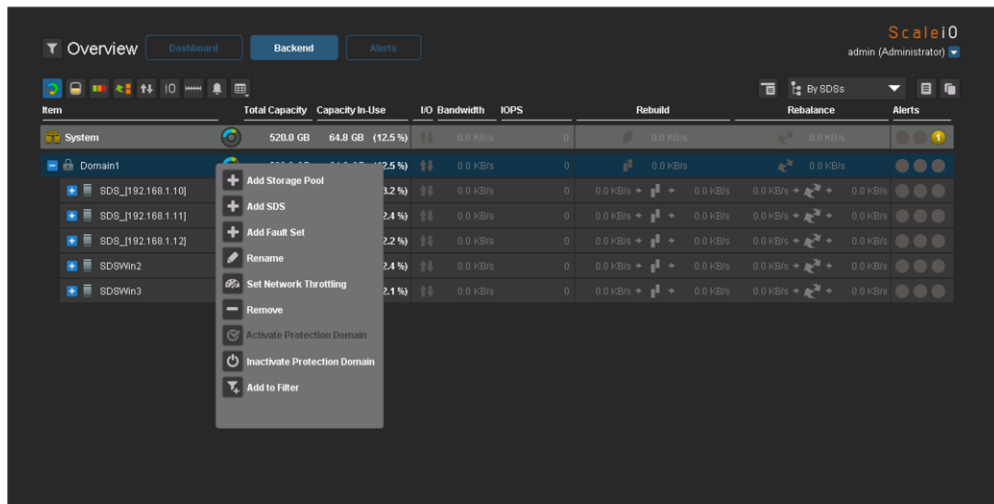
The Preferences drop-down menu in the top right corner displays as a username, for example, “admin,” and has three options: Preferences, Logout, and About. Click Preferences to customize user preferences on the GUI. Here you can specify the number of seconds between screen refreshes and the number of seconds of history to use for calculating reported throughputs, as well as other metrics.

Use Logout to log in as a different ScaleIO user.

Use the About option to get the Installation ID, Software ID, and license details.

You can double-click on a specific Protection Domain or Storage Pool from the drop-down menu in the upper left corner to get dashboard data specific to that Protection Domain or Storage Pool. Depending on the context, the tiles will change to suit the specific context.

ScaleIO GUI - Backend View



The Backend view provides detailed information about components in the system.

The main areas of the Backend view are:

Filter - Allows user to filter the information displayed in the table and property sheets

Toolbar - Allows user to perform an action on the selected row in the table by clicking the appropriate button

Table - Displays detailed information about system components.

Property Sheets - Displays very detailed, read-only information about the component selected in the table. This is useful for fine-grained drill-down actions into specific components within a large cluster without having to resort to the CLI. You can even simultaneously work with multiple Property Sheets, one for each of several related objects – for example, a device, an SDS, a Storage Pool and a Protection Domain.

Note that there are several different selectable Backend views. Shown here is the “Capacity Health” view. Using the drop-down menu you can switch to any of the supported views, such as: Overview, Capacity Usage, Capacity Health, Rebuild and Rebalance, Application I/O, Overall I/O, I/O Bandwidth, and State Summary. These selectable views or presets are designed to suit specific customer use cases.

Within a preset, you can change the hierarchy of items within the table. For example, the current screen shows a view sorted “by SDS” (see upper right drop-down) and items within each SDS. You can also switch the view to show “by Storage Pool”.

Configuration Actions Supported by GUI

- Add/remove Protection Domains
- Add/remove Fault Sets
- Add/remove Storage Pools
- Add/remove SDS
- Configure throttling



Shown here are the configuration actions that can be performed from the Backend view.

ScaleIO vSphere Web Plug-in Main View



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

84

For VMware environments, v1.30 provides a web client plug-in to ease management. The plug-in communicates with the MDM and the vCenter server, enabling users to view components, as well as perform many configuration/provisioning tasks right from within the VMware environment.

The vSphere Web Plug-in can be used to perform initial installation of ScaleIO in a VMware environment, using the “Deploy ScaleIO environment” option as shown on the pane on the right.

ScaleIO vSphere Web Plug-in

- Installed and used for initial ScaleIO cluster deployment
- Can be accessed from home screen on vSphere Web Client
- Basic management activities
 - ▶ Add/remove Protection Domains
 - ▶ Add/remove Storage Pools
 - ▶ Add/remove SDS/SDC
 - ▶ Create/delete volumes
 - ▶ Map volumes to ESXi servers



Shown here are the configuration actions that can be performed with the plug-in.

ScaleIO CLI - Overview

- Available on primary and secondary MDM immediately after cluster installation
- CLI can be accessed using
 - ▶ Linux and VMware - **scli** command
 - ▶ Windows - **C:\Program Files\emc\scaleio\MDM\bin\scli.exe**
 - ▶ Xen - **siocli** command
- General form of CLI
 - ▶ **scli [--mdm_ip <IP>] <command>**
- Supports command completion
- Online help with **--help** option



The CLI enables you to perform all provisioning, maintenance and monitoring activities in ScaleIO. It is installed and available by default on the primary and secondary MDM.

ScaleIO CLI - Command Subgroups

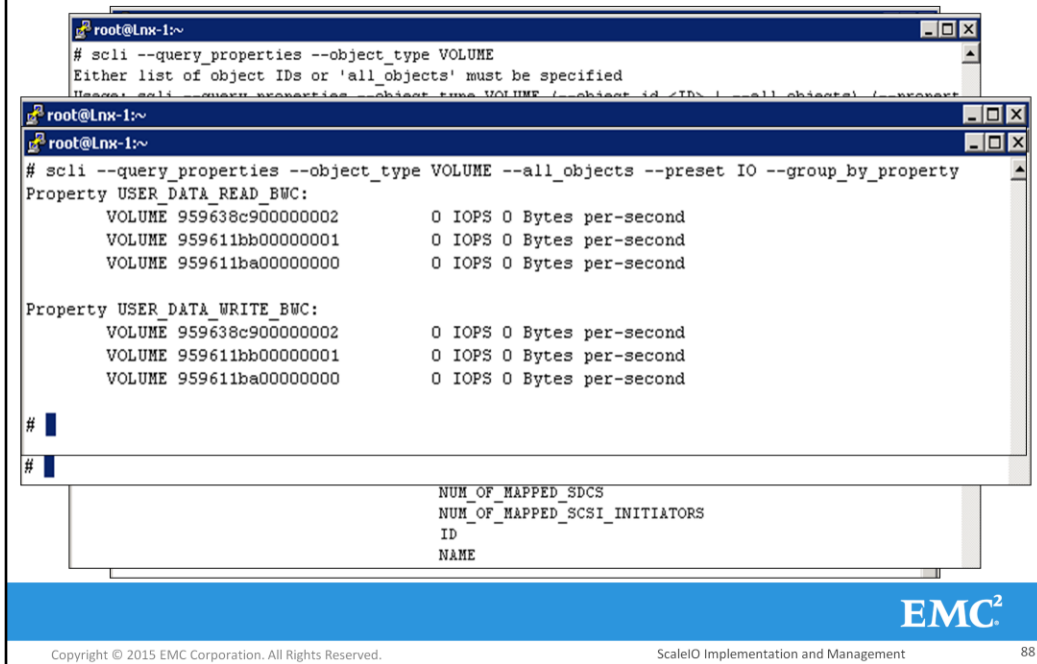
```
[root@LnX-1 ~]# scli --help
Usage: scli --help [Options]
Description: Display help about one or all types of commands
Parameters:
  Options: CHOOSE ONE
    --mdm          Display help about commands relating to the MDM
    --sds          Display help about commands relating to the SDS
    --vol          Display help about commands relating to volumes
    --sdc          Display help about commands relating to SDC
    --SCSI         Display help about commands relating to SCSI
    --general      Display help about general commands, e.g. license, obfuscation, etc.
    --user         Users login and management
    --all          Display help about all commands

[root@LnX-1 ~]#
```



Shown here are the major subgroups of all available ScaleIO CLI commands. For more information about these commands, refer to the online help or ScaleIO User Guide.

Object Types and Object Properties



```
root@lnx-1:~# scli --query_properties --object_type VOLUME
Either list of object IDs or 'all_objects' must be specified
Usage: scli --query_properties --object_type VOLUME [-object_id <ID>] [-all_objects] [-preset]

root@lnx-1:~# scli --query_properties --object_type VOLUME --all_objects --preset IO --group_by_property
Property USER DATA READ_BWC:
  VOLUME 959638c9000000002      0 IOPS 0 Bytes per-second
  VOLUME 959611bb000000001      0 IOPS 0 Bytes per-second
  VOLUME 959611ba000000000      0 IOPS 0 Bytes per-second

Property USER DATA WRITE_BWC:
  VOLUME 959638c9000000002      0 IOPS 0 Bytes per-second
  VOLUME 959611bb000000001      0 IOPS 0 Bytes per-second
  VOLUME 959611ba000000000      0 IOPS 0 Bytes per-second

#
#
```

NUM_OF_MAPPED_SDCS	NUM_OF_MAPPED_SCSI_INITIATORS	ID	NAME
--------------------	-------------------------------	----	------

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved. ScaleIO Implementation and Management 88

ScaleIO defines several object types such as PROTECTION_DOMAIN, STORAGE_POOL, VOLUME etc. You can obtain a complete list of ScaleIO object types using the `--query_properties` command with no arguments.

For each type of object, ScaleIO defines a set of properties. Running the `--query_properties` command with an object type as the only argument produces the complete set of properties for that object type. Note that ScaleIO also defines several presets, or subsets of the full property list, for each object type. This enables the administrator to perform filtering as needed on the command output.

With the `--all_objects` option, the command produces output for all objects in the cluster that belong to the specified type.

You can instead restrict the objects being reported on with a comma-separated list of object IDs in the `--object_id` option.

The `--group_by_property` option provides a different view of the same data. For each property, it shows the value for each object within the specified type.

Managing Protection Domains and Fault Sets

- Protection Domains
 - ▶ Can be added any time
 - ▶ Can use SDSs removed from other Protection Domains
- Fault Sets - Setup
 - ▶ Assigning SDS to Fault Set done only when adding SDS to system
 - ▶ Moving SDS between Fault Sets requires removing SDS from system and adding it back
- Minimal Configuration
 - ▶ Allocating volume requires three devices from different Fault Sets/SDSs



Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

89

ScaleIO allows for reconfiguration of servers across Protection Domains to accommodate dynamic data center environments with constantly changing workloads. Moving an SDS to a new Protection Domain, or reallocating it to a different Fault Set will require first removing the SDS and then adding it back to the ScaleIO system. This can be done without loss of access to any existing ScaleIO volumes.

To ensure protection, the smallest supportable ScaleIO cluster consists of three SDS nodes. A volume provisioned from such a system can survive planned or unplanned shutdown of any one of these nodes.

Protection Domain – Inactivate/Activate

- Commands to activate and inactivate Protection Domain
 - ▶ For maintenance operations
 - ▶ Prevents ScaleIO from automatically rebalancing data
- When “Inactivate Protection Domain” is issued, ScaleIO will:
 - ▶ Determine if any rebuild/rebalance in progress
 - ▶▶ If in progress, delay shutdown until rebuild/rebalance completes
 - ▶▶ Exception when “force shutdown” specified
 - ▶ Block future rebuild/rebalance
 - ▶ Quiesce application I/O and disable access to volumes
 - ▶ Moves DRL mode of all SDSs in Protection Domain to “harden”



Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

90

Users can simply and effectively shut down an entire Protection Domain, thus preventing an unnecessary rebuild/rebalance operation in the case of planned shutdowns of servers. The “Inactivate_Protection_Domain” and “Activate_Protection_Domain” commands serve this purpose.

An example of housekeeping that needs to happen when an SDS goes down for a planned shutdown is the “Dirty Region Log”, an in-memory structure that the SDS maintains in shared memory for consistency of writes, needs to be persisted to disk first. If not, the bitmap is lost upon a reboot. When this happens, the bitmap needs to be rebuilt when the server comes back up, which could be time-consuming. (Note: the DRL needs to get updated each time the SDS does a write. It protects us from a server failure – it is what enables the user to know after the fact that a write was in flight when the server went down). While executing an “Inactivate_Protection_Domain” command, ScaleIO automatically sets the DRL mode to “harden” for the affected servers, causing their DRL bitmaps to be saved to persistent storage. This saves the cost of a complete DRL rebuild when the Protection Domain is reactivated.

In addition, “Inactivate_Protection_Domain” consistently sets all affected volumes to disabled for read/write access.

Adding/Removing/Querying Fault Sets

- **--add_fault_set**
 - ▶ Add empty Fault Set to given Protection Domain
- **--add_sds**
 - ▶ Accept optional Fault Set flag
- **--rename_fault_set**
 - ▶ Rename specified Fault Set
- **--query_fault_set**
 - ▶ Query Fault Set information

```
[root@LnX-1 ~]# scli --query_all_fault_sets --protection_domain_name Domain1
Protection Domain has 1 Fault Sets:
Protection Domain ID: b4ac019d00000000 Fault Set: ID: daa7bc0500000000 Name: FS1 SDS Count: 0/n
[root@LnX-1 ~]#
```



Several commands are available that are specific to creating, managing, modifying and removing Fault Sets. Refer to the ScaleIO User Guide for complete documentation and usage of these commands.

Managing Storage Pools - Storage Tiering

Listing storage pools properties:

```
scli --query_storage_pool --protection_domain_name  
<domain_name> --storage_pool_name <pool_name>
```

```
[root@Lnx-1 ~]# scli --query_storage_pool --protection_domain_name Domain1 --storage_pool_name default
Storage Pool default (Id: 74fc5a45000000000) has 4 volumes and 72.0 GB (73728 MB) available for volume allocation
The number of parallel rebuild/rebalance jobs: 2
Rebuild is Enabled and using Limit-Concurrent-IO policy with the following parameters:
  Number of concurrent IOs per device: 1
Rebalance is Enabled and using Favor-Application-IO policy with the following parameters:
  Number of concurrent IOs per device: 1, Bandwidth limit per device: 10240 KB per second
Zero padding is disabled
Spare policy: 26% out of total
RAM cache is used
RAM cache write handling mode is 'passthrough'

316.0 GB (323577 MB) total capacity
168.2 GB (172283 MB) unused capacity
0 Bytes snapshots capacity
65.6 GB (67164 MB) in-use capacity
1.6 GB (1628 MB) thin capacity
65.6 GB (67164 MB) protected capacity
0 Bytes failed capacity

Volumes summary:
  2 thick-provisioned volumes. Total size: 32.0 GB (32768 MB)
  2 thin-provisioned volumes. Total used size: 814.0 MB (833536 KB)
[root@Lnx-1 ~]#
```

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

92

Storage Pools can be configured within each Protection Domain. This is typically done to take advantage of storage tiering by device type. Refer to the User Guide for complete documentation on all commands related to Storage Pools.

ScaleIO Volumes - Thin vs. Thick Provisioning

- Volumes may be thinly provisioned
 - ▶ Size < 1PB
 - ▶ Metadata provisioned according to size
 - ▶ Default Thick
- Chunks allocated as needed
 - ▶ 1MB
- Trim supported
 - ▶ OS may remove unused blocks
 - ▶ 1MB aligned
 - ▶ Zero returned when read
- Events
 - ▶ Capacity thresholds (configurable)



THIN



THICK

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

93

By default, ScaleIO volumes are “thick-provisioned,” or fully allocated. ScaleIO 1.30 introduces the thin provisioning option, which allows ScaleIO volumes to grow on an “as-needed” basis. For thin volumes, capacity is incrementally allocated when needed in 1MB chunks. At volume creation time, no chunks are allocated to ScaleIO thin volumes.

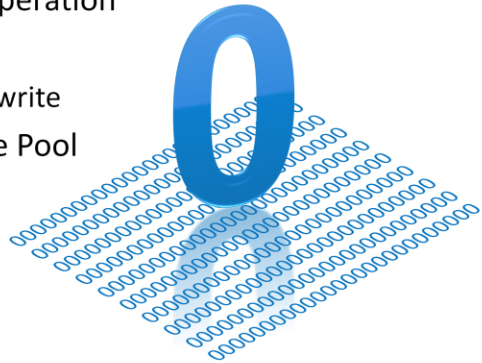
Note that regardless of whether a volume is defined as thick or thin, the metadata for it will be fully provisioned. Neither thin nor thick ScaleIO volumes can exceed 1PB in capacity.

ScaleIO offers “Trim” capability which enables the host to instruct ScaleIO to decommission previously provisioned Blocks in a volume that are no longer in use. This can be done for Blocks which are aligned to 1MB boundaries.

There are two significant capacity thresholds when thin provisioning is used. A lower threshold (percent capacity used) results in a “warning” event being generated. Reaching the higher threshold of allocated capacity results in an “error” event. These thresholds are configurable.

ScaleIO Volumes - Zero Padding

- Erase with zeroes for both thick and thin
- Padding as needed during write operation
- Performance impact
 - ▶ First 8K write to a chunk = 1 MB write
- Enabled/disabled for each Storage Pool
 - ▶ Default disable zero padding



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

94

When a chunk is allocated to a volume, the contents of that chunk are unknown at that point. ScaleIO provides the option of zero-padding the chunk during the first write to that chunk. This applies both to thin volumes and thick volumes.

Without zero-padding, the chunks have unknown contents in all Blocks that are not actually written by the application. If the application reads any Block that it did not previously write, this may cause issues. Zero-padding ensures that such Blocks contains only 0s.

Note that zero-padding involves additional work that ScaleIO must do on behalf of a write. For example, when an 8KB write is initiated to a Block of a volume for the first time, the entire 1MB chunk containing that Block needs to be zero-padded, resulting in 1MB of data to be written.

By default, zero-padding is disabled for all Storage Pools.

Creating and Provisioning ScaleIO Volumes

- Creating ScaleIO volume:

```
[root@lnx-1 ~]# sccli --add_volume --protection_domain_name default --storage_pool_name default --thin_provisioned --volume_name vol2 --size_gb 16
```

- Listing volume:

```
[root@lnx-1 ~]# sccli --query_all_volumes
Query-all-volumes returned 5 volumes
Protection Domain b4ac019d00000000 Name: Domain1
Storage Pool 74fc5a4500000000 Name: default
Volume ID: 037fab7b00000000 Name: vol1 Size: 16.0 GB (16384 MB) Mapped to 1 SDC Thin-provisioned
Volume ID: 037fab7c00000001 Name: vol2 Size: 16.0 GB (16384 MB) Mapped to 1 SDC Thin-provisioned
Volume ID: 037fab7d00000002 Name: vol3 Size: 16.0 GB (16384 MB) Mapped to 1 SDC Thick-provisioned
Volume ID: 037fab7e00000003 Name: vol4 Size: 16.0 GB (16384 MB) Mapped to 1 SDC Thick-provisioned
Volume ID: 037fab7f00000004 Name: vol-10 Size: 16.0 GB (16384 MB) Unmapped Thin-provisioned
```

- Mapping ScaleIO volume:

```
[root@lnx-1 ~]# sccli --map_volume_to_sdc --volume_name vol-10 --sdc_ip 192.168.1.10
Successfully mapped volume vol-10 to SDC 192.168.1.10.
```

- Listing details of an specific volume:

```
[root@lnx-1 ~]# sccli --query_volume --volume_name vol-10
>> Volume ID: 037fab7f00000004 Name: vol-10 Thin-provisioned Size: 16.0 GB (16384 MB)
Capacity in use: 0 Bytes
Trimmed capacity: 0 Bytes
Storage Pool 74fc5a4500000000 Name: default
Protection Domain b4ac019d00000000 Name: Domain1
Creation time: 2014-07-24 13:54:06
Obfuscation is disabled
Read bandwidth: 0 IOPS 0 Bytes per-second
Write bandwidth: 0 IOPS 0 Bytes per-second
Mapped SDC:
SDC ID: 152484ff00000000 IP: 192.168.1.10
```

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

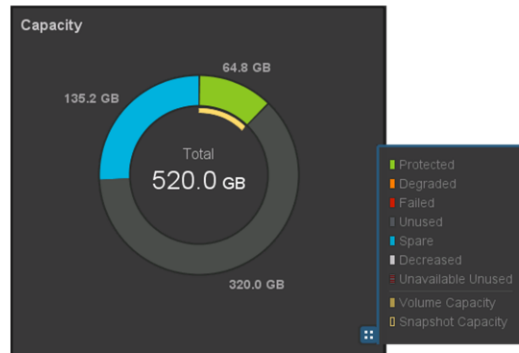
ScaleIO Implementation and Management

95

The CLI includes a comprehensive set of commands related to ScaleIO volume management, including volume creation, volume mapping to SDC clients and volume removal. Shown here are simple examples of these commands. Refer to the User Guide for complete documentation on volume-related commands.

ScaleIO Snapshots - Overview

- Snapshots are volumes
 - ▶ Readable and writable
- Thinly provisioned
 - ▶ Extremely quick creation and deletion (background)
 - ▶ Economical
- Depiction in GUI dashboard
 - ▶ Solid white - volumes
 - ▶ Outlined - snapshots



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

96

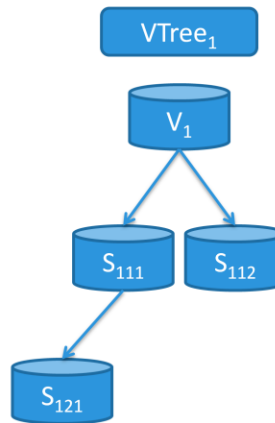
All ScaleIO snapshots are thinly-provisioned, regardless of the form of the original volumes that could be either thin or thick.

The dark-bordered area in the dashboard view of “Capacity” is the capacity used by snapshots. Note that a snapshot can “grow” in one of two ways:

1. The user writes to the original volume. Since the snapshot needs to preserve the original state, it must therefore grow.
2. The user writes directly to the snapshot volume.

Snapshot Organization - VTree

- Root of VTree
 - ▶ Original volume
- Siblings
 - ▶ Snapshots of volume
 - ▶ Support for snapshots of snapshots



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

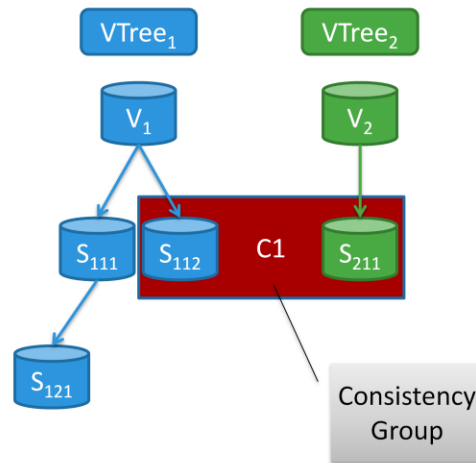
ScaleIO Implementation and Management

97

ScaleIO uses a VTree structure to manage snapshots for volumes. In the example shown, S111 and S112 are snapshots of the original volume, V1, while S121 is a snapshot of the S111 snapshot itself. They are all in the same VTree.

Consistent Snapshots of Multiple Volumes

- Consistency group may be
 - ▶ Several volumes from several Storage Pools
 - ▶ Mixed group of volumes and snapshots
 - ▶ Same volume or snapshot several times
 - ▶ Entire site
- Each group gets unique ID
- S_{112} and S_{211} are consistent



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

98

ScaleIO enables the user to create a consistent snapshot of multiple volumes with a single command. This ensures that write order fidelity is preserved when taking the snapshot, making the snapshots crash-consistent.

In addition, a consistency group can include volumes from multiple Protection Domains. This means you can make a crash-consistent copy of multiple volumes spread all over the data center, if they are all related to the same application. Plus, you can even snapshot the entire data center in a crash-consistent manner, if desired.

The diagram illustrates the Redirect on Write mechanism. At the top, four storage nodes are shown, each containing an SDC (Secondary Data Cache) and an SDS (Secondary Data Store). Below these, four data blocks are represented as stacks of cylinders. The first block has layers labeled S, S, and P. The second block has layers labeled S and P. The third block has layers labeled P and P. The fourth block has a single layer labeled S. A red arrow points from the first block to the third block, indicating a redirect. Below the blocks, a large red arrow points to a central stack of cylinders labeled V_1 . A yellow arrow labeled "WRITE" points from a disk icon to the V_1 stack. A blue arrow points from the V_1 stack to a disk icon labeled S_1 . The text "Redirect on Write" is written in green on the right side of the diagram.

In any case, data is written exactly once. It is all managed within ScaleIO with pointer manipulation.

Snapshot Commands

- Creating a snapshot

```
[root@LnX-1 ~]# scll --snapshot_volume --volume_name vol1 --snapshot_name vol1_snap
Snapshot created successfully
Source volume vol1 => 037fab8000000005 vol1_snap
```

- Map snapshot volume to SDC

```
[root@LnX-1 ~]# scll --map_volume_to_sdc --volume_name vol1_snap --sdc_ip 192.168.1.11
Successfully mapped volume vol1_snap to SDC 192.168.1.11.
[root@LnX-1 ~]#
```

- Creating consistent snapshot

```
[root@LnX-1 ~]# scll --snapshot_volume --volume_name vol2,vol3 --snapshot_name vol2_snap,vol3_snap
Snapshots on 2 volumes created successfully.
Consistency group ID: c4a98f1100000002
Source volume vol2 => 037fab8100000006 vol2_snap
Source volume vol3 => 037fab8200000007 vol3_snap
[root@LnX-1 ~]#
```

- Querying consistent snapshot volume

```
[root@LnX-1 ~]# scll --query_volume --volume_name vol2_snap
>> Volume ID: 037fab8100000006 Name: vol2_snap Snapshot of vol2 Size: 16.0 GB (16384 MB)
Storage Pool 74fc5a4500000000 Name: default
Protection Domain b4ac019d00000000 Name: Domain1
Creation time: 2014-07-28 10:41:40
Obfuscation is disabled
Read bandwidth: 0 IOPS 0 Bytes per-second
Write bandwidth: 0 IOPS 0 Bytes per-second
Consistency Group ID: c4a98f1100000002 Volume is unmapped
[root@LnX-1 ~]#
```

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

100

Shown here are some examples of ScaleIO commands for working with snapshots and consistency groups.

Role-based Access - Overview

- ScaleIO requires authentication to operate
- Supports multiple users
 - ▶ “admin” user is created at installation time by default
- User may have one of several defined roles
- Authentication required at start of session
 - ▶ Multiple operations can be run successively within session
- Active Directory/LDAP integration not in place yet



ScaleIO version 1.30 introduces role-based access that allows the capability of defining multiple local users on a ScaleIO system with configurable roles. Active Directory and LDAP integration are on the roadmap.

Authenticating once is sufficient to initiate a session within ScaleIO from which multiple operations can be performed without re-authenticating for each operation.

Configurable Roles for ScaleIO Users

- Monitor
 - ▶ Allows querying system information
- Configure
 - ▶ Additionally allows configuring the system
- Administrator
 - ▶ Additionally allows users management

```
[root@LnX-1 ~]# scli --query_users
System has 5 users
ID                Role      Need-New-Pass  User Name
-----
5805365f00000004 Administrator Yes           ganesh
5805365e00000003 Monitor      Yes           steph
5805365d00000002 Configure   Yes           lee
5805365c00000001 Configure   Yes           raph
5805365b00000000 Administrator No            admin
[root@LnX-1 ~]#
```

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

102

Each ScaleIO user has one of three different roles: Monitor, Configure, or Administrator.

The “Monitor” role is query-only.

The “Configure” role allows the user to make changes to the ScaleIO system and query.

The “Administrator” role includes the above capabilities, plus the ability to change the user configuration itself, for example, to add or remove users. It is possible to have several users with the “administrator” role.

ScaleIO User Sessions

- Authenticate once, perform many operations per session
- Session ends due to any of these conditions
 - ▶ Ten minutes of inactivity
 - ▶ Eight hours after login
 - ▶ When user logs out

```
[root@LnX-1 ~]# scli --login --username admin --password P@ssw0rd
Logged in. User role is Administrator
[root@LnX-1 ~]# scli --query_all_sds
Query-all-SDS returned 4 SDS nodes.

Protection Domain b4ac019d00000000 Name: Domain1
SDS ID: 580bedbc000000003 Name: SDSWin2 State: Connected, Joined IP: 192.168.1.21 Port: 7072
SDS ID: 580bedbb000000002 Name: SDS_[192.168.1.12] State: Connected, Joined IP: 192.168.1.12 Port: 7072
SDS ID: 580bedba000000001 Name: SDS_[192.168.1.11] State: Connected, Joined IP: 192.168.1.11 Port: 7072
SDS ID: 580bedb9000000000 Name: SDS_[192.168.1.10] State: Connected, Joined IP: 192.168.1.10 Port: 7072
```



When a user is authenticated by the system, all commands performed will be done with their respective role until a logout is performed, or until the session expires, by reaching one of the following timeouts:

- Maximum session length (default: 8 hours)
- Session idle time (default: 10 minutes)

Restricted SDC Access

- Capability to prevent SDC access to a ScaleIO cluster until it is explicitly added
- When restriction is enabled:
 - ▶ Must first grant access to the SDC with the `--add_sdc` command, before any volumes can be mapped to it
 - ▶ SDCs can be identified by either IP address or hostname (if DNS is configured)
 - ▶ SDCs can be granted access before they are installed
 - ▶ Before granting access: `--map_volume_to_sdc` command will fail for an SDC
- New CLI commands:
 - `--set_restricted_sdc_mode <enabled/disabled>`
 - `--query_restricted_sdc_mode`
 - `--add_sdc --sdc_ip <IP address | hostname>`



For improved access control, Version 1.31 introduces a feature to restrict access to ScaleIO volumes to an explicit set of SDCs. This is referred to as the restricted SDC access feature. When the restriction is in place, the administrator is required to explicitly grant access to an SDC before any volume-related operations specific to that SDC can be performed – for example, mapping volumes to the SDC.

Several new commands have been implemented to support this feature, including `query_restricted_sdc_mode` and `set_restricted_sdc_mode` to examine and set the state of restriction, and `add_sdc` to grant SDC access. Note that `add_sdc` can be run for a host before the SDC is installed on it.

Configuring Hot Standby MDM

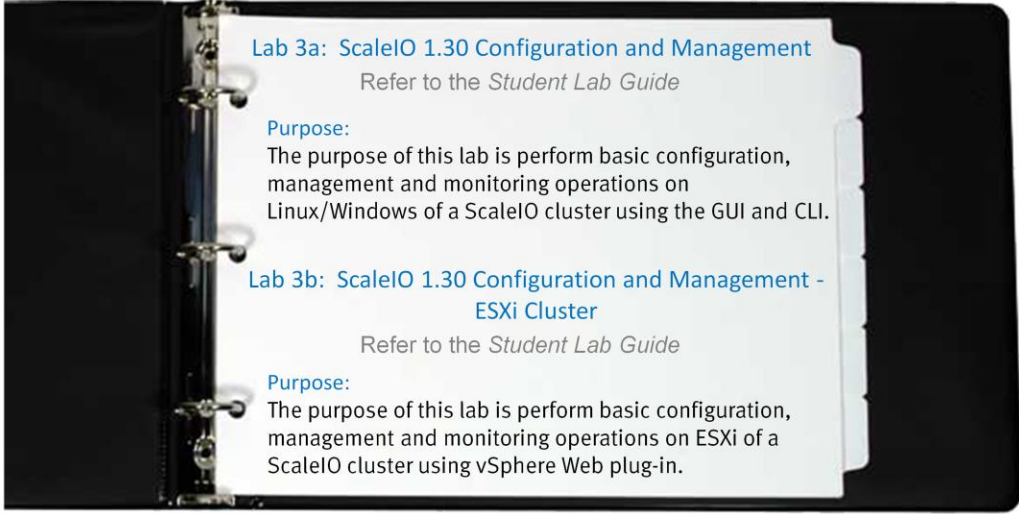
- V1.30 supports up to eight MDMs and up to two tie-breakers
 - ▶ Includes two active MDMs and one active tie-breaker
- When ScaleIO identifies MDM cluster state is permanently broken, it phases in the standby MDM or tie-breaker instance
- Feature minimizes vulnerability of metadata when MDM fails
- Hot standby requires manual installation to meet specific customer request
 - ▶ Configuration guidance will be available from EMC Support



The “hot standby” feature was added in v1.30 to address the customer concern that when one of the two MDMs goes down, the entire cluster is now operating with just one available copy of the metadata. The ability to add a standby MDM at this point minimizes the risk of losing metadata changes going forward.

In general, up to eight MDMs (including the two active MDMs) and two tie-breakers can be configured per cluster. To configure and activate a standby MDM to meet a specific customer request, a scripted solution is available. Please contact EMC Support for more information.

Lab Exercise



Lab 3a: ScaleIO 1.30 Configuration and Management
Refer to the *Student Lab Guide*

Purpose:
The purpose of this lab is perform basic configuration, management and monitoring operations on Linux/Windows of a ScaleIO cluster using the GUI and CLI.

Lab 3b: ScaleIO 1.30 Configuration and Management - ESXi Cluster
Refer to the *Student Lab Guide*

Purpose:
The purpose of this lab is perform basic configuration, management and monitoring operations on ESXi of a ScaleIO cluster using vSphere Web plug-in.

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved. ScaleIO Implementation and Management 106

Please proceed to Lab Exercises 3a and 3b, ScaleIO 1.30 Configuration and Management and ScaleIO 1.30 Configuration and Management - ESXi Cluster.

Module 4: Monitoring, Performance and Troubleshooting

Upon completion of this module, you should be able to:

- Monitor the status and health of a production ScaleIO cluster using all available interfaces
- Perform advanced configuration and performance tuning of a functional ScaleIO cluster
- Identify, troubleshoot and correct basic issues in a ScaleIO environment



This module focuses on monitoring various aspects of a ScaleIO cluster. It also covers advanced configuration capabilities that are provided for performance tuning in production environments. Basic troubleshooting information for the V1.30 version is also included.

Monitoring ScaleIO using the GUI (Live Demo)

- GUI monitoring capabilities will be covered using a live demonstration:
 - ▶ **Dashboard, Backend** and **Alerts** views
- In the **Backend** view:
 - ▶ Working with the **Table** and its contents
 - ▶ Changing what is displayed using the **Drop-down** menu
 - ▶ Using the **Filter** and **SDS/Storage Pools** selector
 - ▶ Using the **Toolbar** to perform actions on a selected object
 - ▶ Displaying the **Property Sheet** for a selected object



Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

108

At the top-most level, the GUI provides three major selectors for types of views available: the **Dashboard, Backend** and **Alerts** selector buttons.

The Backend view provides detailed information about components in the system.

The main areas of the Backend view are:

- **Table** - Displays detailed information about system components, and a wide range of information that is accessed from the Property Sheets
- **Drop-down menu** – enables the component types and metrics that are displayed in the Table
- **Filter** - Allows user to filter the information displayed in the table and property sheets
- **Toolbar** - Allows user to perform an action on the selected row in the table by clicking the appropriate button
- **Property Sheets** - Display very detailed read-only information about the component selected in the table. This is useful for fine-grained drill-down into specific components within a large cluster without having to resort to the CLI. Users can even simultaneously work with multiple Property Sheets, one for each of several related objects, such as a device, an SDS, Storage Pool and Protection Domain.

Segregation of Network Traffic - IP Roles

- Roles can optionally be used to isolate specific types of ScaleIO traffic to specific IP subnets
- Each SDS may have up to eight IP addresses, each with one of three roles:
 - ▶ SDC_ONLY: For front-end application I/O to ScaleIO volumes
 - ▶ SDS_ONLY: For MDM control traffic, RAID copy, rebuild/rebalance
 - ▶ ALL: For all types of ScaleIO traffic (default)
- Configured using `--sds_ip_role` option when adding SDS
 - ▶ `--sds_ip` and `--sds_ip_role` can both be comma-separated lists
- ScaleIO validates that each SDS has at least one IP address to service SDC and SDS roles



Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

109

ScaleIO allows for use of multiple IP addresses on each SDS to segregate the different types of network traffic in the system. This is done using the “IP Roles” feature that applies specifically to the SDS servers. At SDS add time, users can specify a list of up to eight IP addresses for the SDS, and using the `-sds_ip_role` option and specify a role for each IP address, as well.

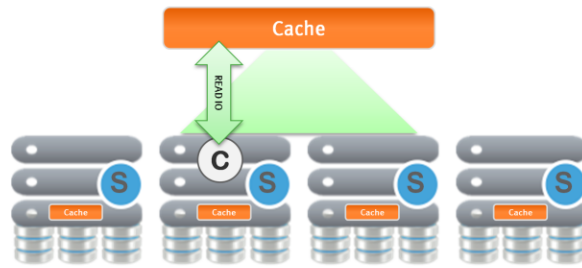
The “SDC_ONLY” role restricts use of the SDS IP address to front-end application traffic only (SDC to SDS, to service read or write, to ScaleIO volume).

The “SDS_ONLY” role is used for MDM control traffic, RAID copy, rebuild and rebalance traffic.

By default, every IP address is configured with the “ALL” role, which means it can be used for all traffic.

ScaleIO monitors the system continuously for availability of an IP address on each SDS to perform both the SDC and SDS roles. Note this need can be met with a single IP address with the “ALL” role.

Read RAM Cache



- Benefit: accelerates READ operations to ScaleIO volumes
- Write Mode: can be set to Store or Passthrough (invalidate)
- Cache situated in each SDS: distributed cache architecture
 - ▶ Cache size: 128MB-128GB per SDS; can be changed as needed
 - ▶ Application I/O Blocks of up to 128K cached
 - ▶ Rebalance/rebuild I/O is **not** cached

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

110

ScaleIO 1.30 introduces the Read RAM Cache, which is situated in the RAM of the SDS servers between the SDS software component and the disks that were contributed to ScaleIO. The purpose is to speed up read operations, particularly to magnetic media. ScaleIO aggregates the RAM contributed by each SDS server (128 to 128 GB *per server*) and uses a distributed cache paradigm.

On a given server, while resizing the cache size dynamically is supported, note this results in a cache purge and rebuild.

For write operations, the cache behavior can be set to either “store” or “writethrough”. In “Store” mode, the new data is written to both the device and to the cache. In “Passthrough” mode, the new data is written to the device only. The cache location is invalidated.

Read RAM Cache - Configuration and Operations

- Can be configured for each Storage Pool and each SDS
- For each Storage Pool:
 - ▶ Use/Don't Use (default: Use)
 - ▶ Write Mode: Passthrough or Store (default: Passthrough)
- For each SDS:
 - ▶ Enabled/Disabled (default: Enabled)
 - ▶ Size (default: 128MB)
 - ▶ Freeze
 - ▶ Purge
- Purge, Disabled, and Resize cause all cache contents to be cleared



Shown here are the configuration options and the various operations supported for Read RAM Cache.

Note that the Purge, Disable and Resize operations will all cause the current cache contents to be cleared immediately.

ScaleIO Throttling - Objectives

- Mechanisms to limit bandwidth used by:
 - ▶ Application I/O to ScaleIO volumes
 - ▶ Rebuild operations
 - ▶ Rebalance operations
- Prioritization: rebuild/rebalance vs application IO
 - ▶ Manage impact from rebuild/rebalance on application IO
- Regulate network load generated by ScaleIO
 - ▶ Control impact from all ScaleIO traffic on other data center traffic

Note: Engagement with L3 or L4 support is required for advanced throttling customizations



Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

112

With version 1.30, ScaleIO offers several mechanisms for the administrator to control the bandwidth used by application I/O activity, rebuild and rebalance operations. These mechanisms provide the capability to control three different types of ScaleIO traffic: application I/O, ScaleIO rebuild operations and rebalance operations. There are two possible reasons for doing this. The first is to ensure that application I/O performance does not degrade to unacceptable levels due to rebuild or rebalance. The second is to limit the impact of ScaleIO itself on the data networks, which may need to be shared with other traffic.

In general, ScaleIO is designed to work “out of the box” and should not require significant changes to default behavior for proper operation. Throttling customizations should therefore not be attempted without a holistic understanding of the specific data center environment and the infrastructure supporting ScaleIO. It is expected that L3 or L4 support from EMC will be involved in the customization process.

Let’s explore four different ways in which throttling can be configured with ScaleIO 1.30.

Rate Limits - By Type of I/O Operation

- For each Protection Domain, these MB/sec limits can be individually configured:
 - ▶ Maximum rebuild rate
 - ▶ Maximum rebalance rate
 - ▶ Total network bandwidth used by ScaleIO
 - ▶▶ Includes rebuild, rebalance and application I/O



First, the following set of limits can be configured for each Protection Domain:

- Maximum rebuild rate (MB/sec)
- Maximum rebalance rate (MB/sec)
- Total network bandwidth used by ScaleIO (application IO + rebuild/rebalance)

Note that when both rebuild and rebalance occur simultaneously, the aggregate bandwidth consumed by both will not exceed the individual maximum for each type. For example, if the rebuild rate is limited to 60MB/sec, and the rebalance rate to 30MB/sec, then concurrently running rebuild jobs and rebalance jobs is limited to 60MB/sec.

Disabling and Enabling Rebuild/Rebalance

- *For each Storage Pool*, disable or enable either one or both:
 - ▶ Rebuild
 - ▶ Rebalance
- Motivation: to defer rebuild/rebalance to off-peak hours



Second, it is also possible to disable and enable the rebuild/rebalance. This control applies to a specific Storage Pool.

A practical example of usage is when new servers are added to the cluster during the work week when application I/O workload is high. To avoid additional network congestion from rebuild or rebalance operations, they can be deferred to the weekend. The decision to defer rebuilds should be carefully considered, since a rebuild fixes a protection issue (single copy of data). Deferring rebalance, but not rebuilds, may make more sense in a production environment.

Concurrency Limit - Rebuild/Rebalance

- *For each Storage Pool*, specify maximum number of concurrent jobs of type: rebuild/rebalance
- Default value: three jobs
- Motivation: to allow faster devices (SSDs) to process more I/O in parallel
 - ▶ Can be increased up to 20



Third, it is possible to limit the number of simultaneously active rebuild/rebalance jobs. This control also applies to a specific pool, which is appropriate, since it should depend on the capabilities of the specific device type. For example, SSD devices can typically handle more IO threads in parallel than magnetic disk drives. The default value for maximum jobs is three. It can be set to as high as 20 for a specific Storage Pool.

Priority Setting - Application vs Rebuild/Rebalance

- For each Storage Pool, configure one of four priority schemes for application I/O vs rebuild and application I/O vs rebalance:
 - ▶ No priority
 - ▶ Serialize (single-thread) all rebuild/rebalance operations
 - ▶ Favor application I/O
 - ▶ Dynamic bandwidth throttling



Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

116

Fourth, it is possible to define priorities for application IO, rebuild and rebalance workloads. This is done by configuring one of four different settings for prioritization scheme:

- No priority
- Serialize (single-thread) - all rebuild/rebalance operations
- Favor application I/Os - will actually limit the number of IOPs due to rebuild and rebalance
- Dynamic bandwidth throttling - both IOPs and bandwidth limits are first defined for rebuild and rebalance, and whenever application IOPs exceed a certain internally-configured value within ScaleIO, the throttling goes into effect; otherwise, application load is considered light and no throttling is done

The administrator can choose one of these four prioritization schemes for each Storage Pool. For a given pool, a different prioritization scheme can be set for rebuild and for rebalance.

Configuring QoS for Volumes

- Throughput limit (IOPS) per ScaleIO volume
 - ▶ Minimum settable value: 10 IOPS; default: 0 (unlimited)
- Bandwidth limit (MB/sec) per ScaleIO Volume
 - ▶ Minimum settable value: 1 MB/sec; default: 0 (unlimited)
- Either limit, or both, can be specified for a SDC/volume pair
 - ▶ *Volume needs to be mapped to an SDC before limits can be set*
- Use the **--set_sdc_volume_limits** command:

```
# scli --set_sdc_volume_limits --volume_name host10_vol3 --sdc_ip  
192.168.1.10 --limit_iops 20 --limit_bandwidth 1  
Successfully modified limits from SDC 192.168.1.10 to volume  
host10_vol3:IOPS limit set to 20.Bandwidth limit set to 1 MB/sec
```



ScaleIO enables the storage administrator to limit either the IOPs, or the bandwidth in MB/sec, for any given volume. This brings in the capability to define multiple block storage tiers based on the quality of service (QoS).

To specify either limit, the volume needs to be mapped to an SDC. The limit can then be specified using the `scli --set_sdc_volume_limits` command by providing both a volume identifier and an SDC identifier.

Background Device Scanner

- Background Verification of data integrity
- Discover issues with cold data
- Two Modes
 - ▶ Device Only
 - ▶ Data Comparison



- Error Generated: **SCANNER_COMPARE_REPORT**

```
Background device scanner on device ID 2301536800030001 reported compare
error (Device Path: <device path>, SDS: <SDS name and ID>, Peer Device
Path: <peer device path>, Peer SDS: <peer SDS name and ID>, Volume name:
<volume name>, Volume offset: <volume offset>)
```

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

118

The background device scanner scans devices in the system to check for errors.

Scanner mode of operation:

- **device_only:** Perform read operations. Fix from peer on errors
- **data_comparison:** Perform the **device_only** test, and compare the data content with peer

Information about errors will be provided in event reports

```
scli --enable_background_device_scanner  
--disable_background_device_scanner
```

- Protection Domain
 - ▶ Default is All
- Storage Pool
 - ▶ Default is All
- Mode
 - ▶ Device vs compare
- Bandwidth limit
 - ▶ KB/sec (per device)
 - ▶ From 10KB – 10MB (default is 1MB)



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

119

Note: High bandwidth should be used very carefully, and for extreme cases only (such as an urgent need to check some devices), as it may create negative impact on system performance. Setting the background device scanner bandwidth should take into account maximum bandwidth of the devices.

ScaleIO Events - Overview

- ScaleIO generates **events** in response to changes in system during normal operation
- Each event has **severity level**, which can be one of:
 - ▶ INFO
 - ▶ WARNING
 - ▶ Alerts to possible failures that may actually be planned actions
 - Example: SDC disconnects as a result of a planned shutdown
 - ▶ ERROR
 - ▶ System is operational, but some data may be unprotected
 - Example: During recovery phase right after one SDS goes offline
 - ▶ CRITICAL
 - ▶ Some data is unavailable
 - Example: Simultaneous failure of several SDS



Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

120

The ScaleIO system generates and records events in specific response to changing conditions within the cluster.

Every ScaleIO event has a severity level associated with it that ranges from INFO to CRITICAL, depending on the condition causing the event to be raised.

“Critical” events are those that definitely require user intervention; they indicate a data unavailability condition that the ScaleIO cluster cannot recover from without explicit user actions for recovery.

“Warning” and “Error” events may be just transient conditions, or they may require user intervention.

Viewing Events Locally on Primary MDM

- Run **showevents.py** command:

```
[root@LnX-1 ~]# /opt/emc/scaleio/mdm/bin/showevents.py --help

Usage: showevents.py [options]

Options:
  -h, --help                show this help message and exit
  --min_severity=MIN_SEVERITY
                           Specify minimal severity
  --severity=SEVERITY       Specify severity
  --from_id=ID_FROM        Limit by ID
  --to_id=ID_TO            Limit by ID
  --from_date=DATE_FROM    Limit by from Date. Date format is same as it shown in
                           event log (YYYY-MM-DD HH:MM:SS) For example January
                           1st 2013 09:00PM is 2013-01-01 09:00:00
  --to_date=DATE_TO        Limit by to Date. Date format is same as it shown in
                           event log (YYYY-MM-DD HH:MM:SS) For example January
                           1st 2013 09:00PM is 2013-01-01 09:00:00
  -g GREP, --grep=GREP     Search specific word
  --full                   Show full event
```



Events may be viewed when logged in on the primary MDM using the `showevents.py` script provided as part of the ScaleIO installation. Shown here is the command usage and options for this command.

The MDM stores the events in a persistent, private database file, and also periodically archives them.

Event Record Structure

- Every ScaleIO event record has **six** fields:

```
87 2014-08-15 13:51:10.256 CLI_COMMAND_FAILED WARNING  
Command map_volume_to_sdc was not successful. Error code: Could not fi  
nd the SDC  
[root@LnX-1 ~]#
```

- ▶ ID
- ▶ Date
- ▶ Name
- ▶ Severity
- ▶ Message
- ▶ Extended

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

122

Shown here is the structure of any ScaleIO event as recorded in the system. Every ScaleIO event has six distinct fields: ID, Date, Name, Severity, Message and Extended. These fields are highlighted for a particular event as displayed by the `showevents.py` command.

As shown here, the `showevents.py` output produces one line of output per event. Each line uses a color-coded font, based on the severity level for the particular event.

Recommended Action for ScaleIO Event

Look up event by **Name** in User Guide:

SDC Disconnected

Parameter	Description
Name	SDC_DISCONNECTED
Message	SDC (IP: X; ID: Y; GUID: Z) disconnected
Severity	Warning
Description	SDC (IP: X; ID: Y; GUID: Z) has disconnected from the MDM.
Action	Make sure it is expected otherwise this might be a hardware malfunction

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

123

The ScaleIO User Guide documents every possible event by its uniquely identifying Name field. For each type of event, it indicates the recommended action. This example shows the recommended action documented in the User Guide for the event “SDC_DISCONNECTED”.

Forwarding ScaleIO Events to Remote Syslog Server

- Events can be forwarded to local or to remote syslog server with **scli --start_remote_syslog** command
 - ▶ Mandatory parameter: Server IP address
 - ▶ Optional parameters: TCP port number, syslog facility number

Example:

```
scli --start_remote_syslog --remote_syslog_server_ip  
192.168.1.103 --remote_syslog_server_port 514
```

- Stop logging at any time with **--stop_remote_syslog**



ScaleIO events may optionally be forwarded to a remote syslog server. This allows for convenient monitoring in data centers that have standardized on syslog as a mechanism to aggregate logs from all applications.

Call Home - Event Notification by Email

- Can be configured as needed during installation
- *Required*: Email address to which event notifications are sent
- *Optional*: Specific minimum severity (such as “WARNING”)
 - ▶ Default: Send notifications for all events with maximum verbosity



An email address can optionally be configured during ScaleIO installation to receive event notifications automatically.

This configuration process will allow for filtering for a minimum severity level. By default, every event (including “INFO” events) will trigger email notification.

Lab Exercise

Lab 4: ScaleIO Monitoring and Performance Tuning

Refer to the *Student Lab Guide*

Purpose:

Perform various operations using the ScaleIO GUI and the CLI to monitor and tune a ScaleIO storage environment.

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

126

Please proceed to Lab Exercise 4, ScaleIO Monitoring and Performance Tuning.

Module 5: ScaleIO REST API and OpenStack Integration

Upon completion of this module, you should be able to:

- Explain the ScaleIO REST API functionality
- Demonstrate the REST API using simple GET and POST calls
- Describe how ScaleIO can be integrated into an operational OpenStack environment
- Perform and validate various aspects of OpenStack integration



This module focuses on two major external integration points that ScaleIO 1.30 introduces, the REST API, and ScaleIO Cinder Driver for OpenStack. These features enable ScaleIO provisioning, management and monitoring using third-party management stacks external to ScaleIO.

ScaleIO with RecoverPoint Overview

- RecoverPoint v4.1.1_SIO enables replication of ScaleIO LUNs
- Requires a ScaleIO host-based splitter and virtual RecoverPoint Appliances (vRPAs)
- Benefits
 - ▶ All RP recovery features are supported, providing:
 - ▶▶ Any point-in-time recovery
 - ▶▶ Lower TCO
 - ▶▶ Increased user agility with virtual appliance deployment



ScaleIO v1.31.1 and RecoverPoint v4.1.1_SIO enables the replication of ScaleIO LUNs both locally and remotely to other ScaleIO clusters. This disaster recovery solution requires a host-based splitter to be installed on all ScaleIO Data Client (SDC) servers. On the RecoverPoint side for this release, only Virtual RecoverPoint Appliances (vRPAs) are supported.

This solution supports all RecoverPoint features and enables any PiT data recovery, lower TCO and increases the user agility by the use of the virtual RecoverPoint appliance.

RecoverPoint Replication Overview

- Production data available during replication
- Sends only changed data over the wire
- Multiple copies at multiple sites
- Supports multiple splitter types: VNX, VMAX, VPLEX, ScaleIO etc.
- Application-consistent replication
- Server consistent replication
 - ▶ Supports federated collection of servers and storage arrays
- Synchronous and asynchronous data recovery
 - ▶ Any copy can be made available as read/write
 - ▶ Changes to the copy can be incrementally reapplied to the primary on failback



Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

129

EMC RecoverPoint is an enterprise-scale solution designed to protect application data on heterogeneous SAN-attached servers and storage arrays. RecoverPoint provides many features that make it a unique and tested solution for backup and recovery.

RecoverPoint allows for the access of a PiT image, either locally or at another site, while still performing replication and without interrupting production activities. The ability to access data from a copy allows for testing without sacrificing protection. Point in time copies can be created for each write, or the user can choose the amount of data lag that can be tolerated for an application. This option is configurable for each group of volumes and can be edited at any time. RecoverPoint also supports many different splitters for access to storage from VNX, VMAX, and VPLEX for 3rd party arrays, and a new ScaleIO host splitter to support the ScaleIO storage devices.

Use Cases for ScaleIO with RecoverPoint

Protect, Replicate, and Recover



Disaster Recovery

Protect end users against outages caused by

- Natural disasters
- Accidents
- Utility outages
- Technical malfunctions



Operational Recovery

Protect end users against outages caused by

- Data corruptions
- Virus attacks
- Operational errors



Tech Refresh Data Migration

Replicate data for

- Tech refresh, system upgrade
- Data center move or expansion

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

130

The right remote-replication solution can limit your exposure to planned and unplanned downtime, enabling nonstop operation. Perhaps you also need to provide your organization with efficient data replication to meet corporate or governmental standards, while still meeting your total-cost-of-ownership requirements. In addition, you need a flexible solution that changes as your needs change.

RecoverPoint can be used to protect, replicate, and recover data from LUNs in ScaleIO environments. Use cases for disaster recovery can be found in scenarios like natural disasters, accidents, utility outages, and technical malfunctions.

Also, using RecoverPoint is very valuable for recovering from daily operational mishaps like data corruptions, virus attacks, and operational errors.

Lastly, RecoverPoint's data replication technology can help during tech refreshes, system upgrades, and data migrations in the case of a data center move or expansion.

ScaleIO REST API

- Why REST?
 - ▶ HTTPS-based, widely adopted, any web browser can be a client
- What is included?
 - ▶ Full functionality API - any operation that can be done with CLI
 - ▶ Includes operations for initial MDM configuration, cluster setup
- Which REST response format is used?
 - ▶ JSON format
 - ▶ Client should be configured to display responses in JSON format
- Are the supported REST calls documented?
 - ▶ Published API documentation ships with product



REST (**R**epresentational **S**tate **T**ransfer) is a widely-adopted API for the management of storage systems. It is a simple HTTPS-based protocol, enabling any standard web browser to function as the client.

In version 1.30, ScaleIO introduces a REST API, which may be used to perform any action that can be performed using the CLI - not only management actions in an operational system, but also the initial configuration of a primary MDM and setup of the MDM cluster. With the ScaleIO REST API, responses are formatted in Javascript Object Notation (JSON). The client application should be suitably configured for this format.

The ScaleIO REST API is published and included as part of the version 1.30 documentation.

REST Gateway

- ScaleIO component that exposes monitoring and provisioning via REST interface over SSL
- Bundled with IM (Windows or Linux)
- Configured with MDM IPs of a specific cluster
 - ▶ Note: One dedicated REST gateway required for each MDM cluster



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

132

REST API calls can be made to a ScaleIO cluster by accessing the REST gateway of that cluster. The gateway exposes ScaleIO monitoring and provisioning functionality via the REST interface over an HTTPS connection (Secure Sockets Layer or SSL).

The ScaleIO version 1.30 IM ships bundled with the REST gateway. Installing IM automatically causes the REST gateway to be installed as well.

Before it can be used, the gateway must be configured with the MDM IP addresses and access credentials for a specific ScaleIO cluster.

While the MDM addresses can be changed later using a configuration file or via the API itself, it is critical to note that a gateway can only connect to one cluster. In a data center with multiple ScaleIO clusters, each cluster requiring REST access will need its own REST gateway.

ScaleIO REST API Calls - Structure

- Major types defined
 - ▶ **System, ProtectionDomain, SDS, StoragePool, Device**
- GET all objects of given type
 - ▶ /api/types/{**type**}/instances
- GET by ID
 - ▶ /api/instances/{**type::id**}
- POST special action on object
 - ▶ /api/instances/{**type::id**}/action/{**actionName**}



This is a brief description of the structure of commonly-used ScaleIO REST calls.

A key point to note is that even though ScaleIO allows users to supply friendly names for entities, such as Cluster Instance, Protection Domain, Storage Pool or volume, the syntax requires that the unique ID assigned to that entity in GET or POST API calls referring to that entity be specified.

Browser-based REST Client - Typical Layout

The screenshot shows the RESTClient interface. At the top, there is a menu bar with 'File', 'Authentication', 'Headers', and 'View'. Below this is a 'Request' section. It contains a 'Method' dropdown menu currently set to 'GET', a 'URL' text input field with the value 'http://www.example.com', and a 'SEND' button. Below the 'Request' section is a 'Body' section with a text area for entering the request body. The footer of the application shows the EMC logo, copyright information 'Copyright © 2015 EMC Corporation. All Rights Reserved.', and the page number '134'.

A simple graphical browser-based REST client can be used for initial testing and validation of a functional REST gateway communicating with a ScaleIO cluster. The Firefox REST client add-on is a typical example.

Shown here is the layout of the screen for this particular REST client.

The **Authentication** drop-down is used to enter the admin user credentials for the ScaleIO cluster. After these are entered once, they will be used with all subsequent REST calls from this client until they are changed.

The **Headers** drop-down is used to specify that the output format is JSON.

For every call to be issued, the **Method** should be identified; typically as GET or POST.

The **URL** section is where the REST call itself is entered.

For POST calls only, the **Body** section will need to be completed with the required parameters (IDs for example; this depends on the call).

After entering all required information, the **SEND** button is used to initiate the call.

Example of GET Call to ScaleIO Gateway

The screenshot displays a REST client interface. The top section, titled "[.] Request", contains a "Method" dropdown set to "GET" and a "URL" field with the value "https://localhost:443/api/types/System/instances". Below this, the "Headers" section shows "Content-Type: application/json" and "Authorization: Basic YWVWbWV46UEB...". A green arrow points down to the "[.] Response" section. In the response section, the "Response Body (Highlight)" tab is selected, showing a JSON array with one object. The object contains fields for name, capacity alert thresholds, cluster state, system version, and MDM actor ports. The "name" field is null, and the "mdmClusterState" is "ClusteredNormal".

```
1. [
2.   {
3.     "name": null,
4.     "capacityAlertHighThresholdPercent": 80,
5.     "capacityAlertCriticalThresholdPercent": 90,
6.     "mdmClusterState": "ClusteredNormal",
7.     "systemVersionName": "EMC ScaleIO Version: R1_30
8.     "primaryMdmActorPort": 9011,
9.     "secondaryMdmActorPort": 9011,
```

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

135

Shown here is a simple GET call as issued from the Firefox browser-based client.

Note specifically that the **URL** field must begin with the IP address of the REST gateway. In this case, it happens to be the same host as the one running this REST client.

The response to the GET call may be viewed in the **Response** pane as shown.

Example of POST Call to ScaleIO Gateway

The screenshot displays a REST client interface with the following configuration:

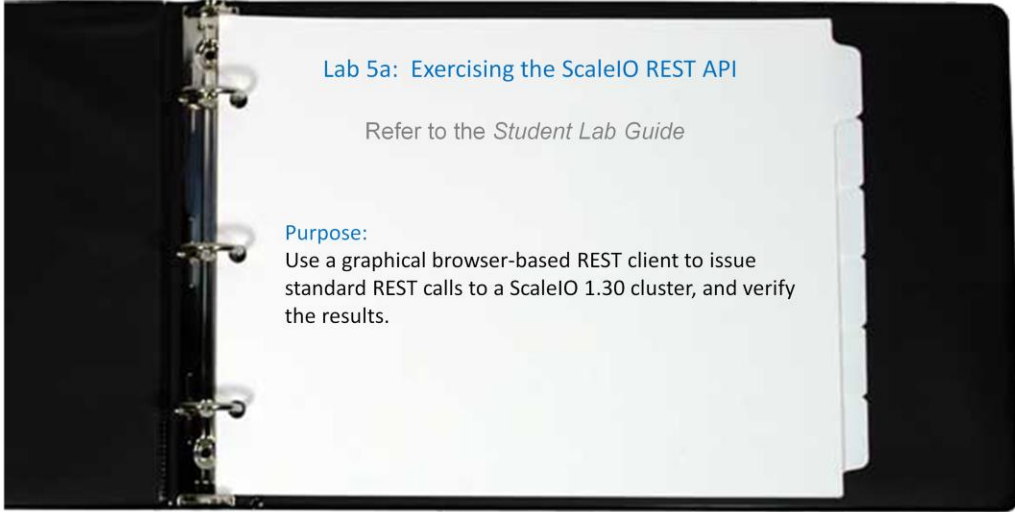
- Method:** POST
- URL:** `https://localhost:443/api/instances/System::5fde8edd5eaf4a27/action/snapshotVolumes`
- Headers:**
 - Content-Type: application/json
 - Authorization: Basic YWRtaV46UEB...
- Body:**

```
{  "snapshotDefs": [    {      "volumeId": "db0a0f7d00000000",      "snapshotName": "SNAP_vol1"    }  ]}
```

The bottom of the interface features the EMC logo and the text "ScaleIO Implementation and Management 136".

In the second example, a POST call is being issued, this time to create a snapshot of a volume. Note that the URL specifies the instance ID, which needs to be obtained first with a prior GET call. The **BODY** section is used as part of this POST call to convey the ID for the source volume and the desired name of the snapshot volume that is created.

Lab Exercises



Lab 5a: Exercising the ScaleIO REST API

Refer to the *Student Lab Guide*

Purpose:
Use a graphical browser-based REST client to issue standard REST calls to a ScaleIO 1.30 cluster, and verify the results.

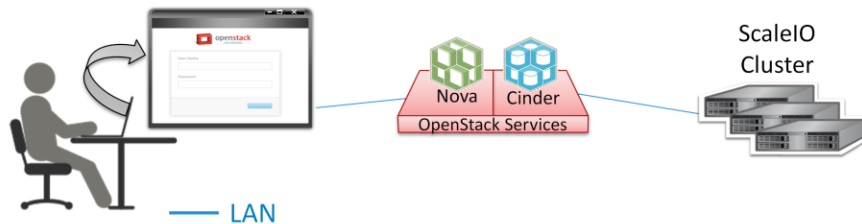
EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved. ScaleIO Implementation and Management 137

Please proceed to Lab Exercise 5a, Exercising the ScaleIO REST API .

ScaleIO Integration with OpenStack

- ScaleIO v1.30 includes a plug-in for OpenStack
 - ▶ Python scripts enabling integration with **Cinder** and **Nova**
- Primary benefit: can provision ScaleIO Block volumes to OpenStack-managed VMs with OpenStack UI or Cinder commands
 - ▶ Cloud administrators do not have to learn native ScaleIO user interfaces for routine storage provisioning or management



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

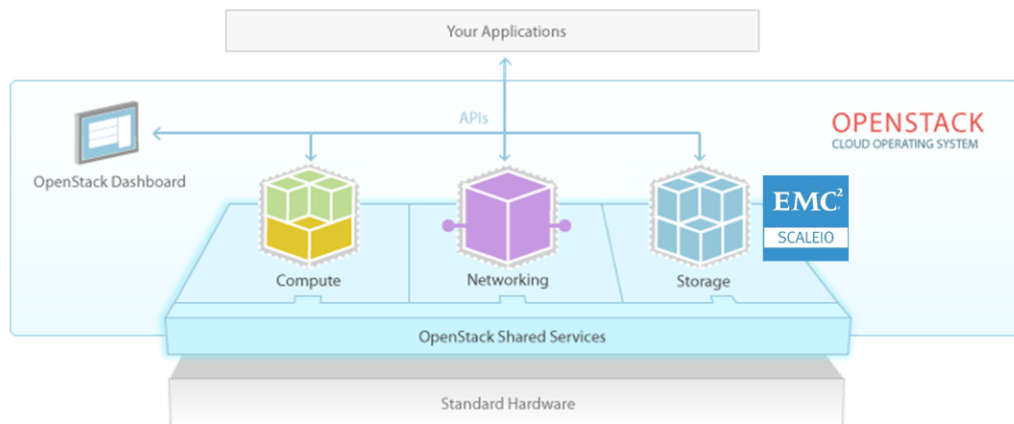
ScaleIO Implementation and Management

138

ScaleIO v1.30 facilitates integration with OpenStack cloud environments by including a plug-in for OpenStack. The plug-in is implemented as Python scripts that need to be installed on OpenStack nodes.

The key benefit of integration is that the cloud administrators can provision Block storage volumes from a ScaleIO cluster directly to OpenStack-managed VMs. And, they can accomplish that using their own familiar OpenStack user interfaces, the Horizon GUI, or Cinder commands.

OpenStack Integration Overview



- ScaleIO Volumes are presented to OpenStack as block devices
- Configured through OpenStack Dashboard
- Commands executed through ScaleIO REST Gateway

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

139

OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a data center. These are all managed using a dashboard or a command line interface that gives administrators control, while allowing their users to provision resources through a web-based interface.

In the case of Block storage, OpenStack provides the Cinder solution, which is a block storage solution for use with servers and applications. Cinder is designed to work with widely available virtualization technologies, bare metal, and high-performance computing configurations. It can integrate with legacy systems and with third-party technologies, such as EMC ScaleIO.

The ScaleIO elastic storage solution includes a Cinder driver, which interfaces between ScaleIO and OpenStack, and presents volumes to OpenStack as block devices which are available for block storage. It also includes an OpenStack Nova driver, for handling compute and instance volume related operations. The ScaleIO driver executes the volume operations by communicating with the backend ScaleIO MDM through the ScaleIO REST Gateway.

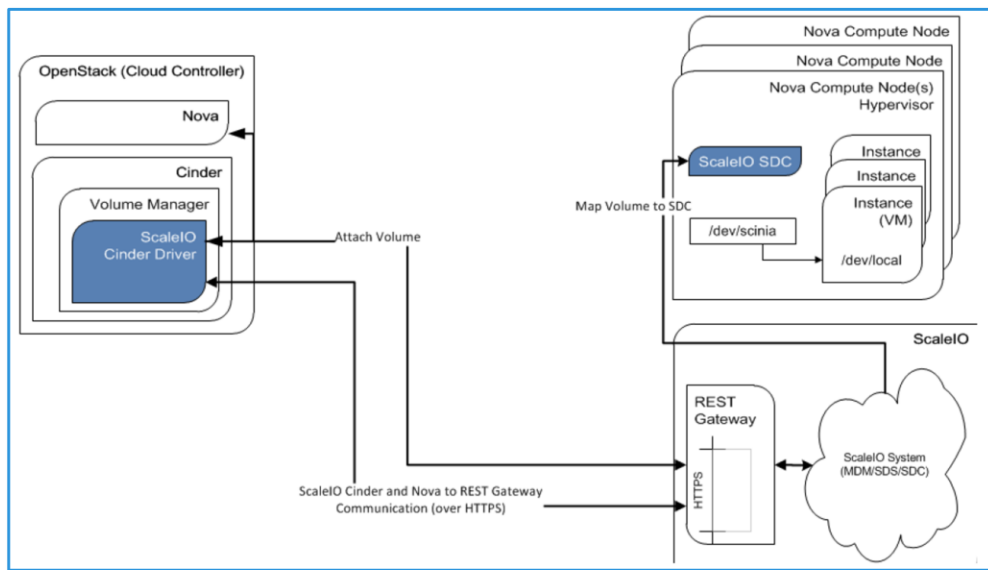
Supported Operations

- Create volume
- Delete volume
- Attach volume
- Detach volume
- Create snapshot
- Delete snapshot
- Create volume from snapshot
- Copy image to volume
- Copy volume to image
- Extend volume



EMC²

OpenStack Integration Architecture



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

141

The following figure shows a typical flow for mapping a volume. The OpenStack flow is as follows:

- Create volume (Cinder operation)—calls the ScaleIO `add_volume` command
- Attach volume to instance—initiated by Nova:
 1. Nova compute manager `attach_volume` is called.
 2. Triggers the Cinder driver to perform the volume attach operation at the storage side (`initialize_connection`)—not needed in the ScaleIO scenario.
 3. Nova LibvirtDriver `attach_volume` is called, which calls the `connect_volume` of the specific LibvirtVolumeDriver (determined by the `volume_driver_type` that the Cinder driver returns in the connection information) to finish the attach operation on the Hypervisor side.

This calls the ScaleIO `map_volume_to_sdc` command and then verifies that the volume is exposed to the hypervisor.

4. After the volume is attached to the compute node, it is passed through to the VM by OpenStack.

OpenStack Integration Requirements

- OpenStack:
 - ▶ Icehouse and Juno releases
 - ▶ All-in-one, or multi-node installation
 - ▶ RHEL 6.x or 7.x
 - ▶ KVM Hypervisor
- ScaleIO Environment
 - ▶ SDC Installed and configured on OpenStack Compute Node
 - ▶ ScaleIO Gateway (REST) installed and accessible
- Restrictions
 - ▶ No support for special chars in Storage Pool or Domain
 - ▶ OpenStack displays volume size as requested (eg 10GB) vs actual (16GB)



Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

142

The following environment is supported:

- Icehouse and Juno releases, with either all-in-one OpenStack installation, or multi-node (Compute nodes) installation
- RHEL 6.x or 7.x (or their CentOS equivalents), using KVM as the hypervisor engine
- The ScaleIO SDC component must be installed and configured on the OpenStack compute node
- The ScaleIO gateway (which includes the REST gateway) must be installed, as part of the ScaleIO deployment. This gateway is typically installed at the beginning of ScaleIO installation. For more information, see the ScaleIO Installation Guide. The ScaleIO gateway must be accessible from the OpenStack cloud controller. Ensure that you know the IP address of the ScaleIO gateway, and its user name and password, because they are required for the Cinder driver's installation and configuration procedures.
- ScaleIO REST Gateway certificate validation—the OpenStack ScaleIO driver communicates with the REST Gateway through https, (over SSL). By default, the driver ignores verification of the ScaleIO REST Gateway's SSL certificate, but it can verify the certificate if the following configuration parameters are defined:
 - **verify_server_certificate**—set to **True**, if the server's certificate must be verified, and to **False** if no verification is required.
 - **server_certificate_path**—If the parameter **verify_server_certificate** is set to **True**, specify the location of the .pem file containing the server's certificate.

For instructions for generating a self-signed certificate using the keytool utility, see the EMC ScaleIO Installation Guide.

- The OpenStack driver does not support Storage Pool and domain names that contain special characters, including URL escape characters such as #, /, and so on. If you would like to create a volume on such a Storage Pool or domain in OpenStack, ensure that you specify the Storage Pool or domain using its ID, and not its name.
- ScaleIO volume size is limited to a basic granularity of 8 GB. If volume size is not a multiple of 8, the size is rounded up. For example: a request to create a volume of 10 GB will create a volume of 16 GB. In OpenStack, the volume will be displayed with its requested size (10) instead of its actual size (16). A corresponding warning message will be printed to the Cinder log.

Manual Installation Steps

On Nova nodes

1. Install /config SDC
2. Copy scaleiolibvirtdriver.py to nova/virt/libvirt
3. Edit /etc/nova/nova.conf
4. Copy scaleio.filters to /etc/nova/rootwrap.d
5. Restart the cinder-volume services

On Cinder nodes

1. Install /config SDC
2. Copy scaleio.py to /cinder/volume/drivers/emc
3. Copy scaleio.filters to /etc/cinder/rootwrap.d
4. Edit cinder.config in /etc/cinder/cinder.conf
5. Create ScaleIO driver configuration file called /etc/cinder/cinder_scaleio.config
6. Restart the cinder-volume services



NOVA

1. Install and configure the SDC component.
2. Copy the file scaleiolibvirtdriver.py to the nova/virt/libvirt directory.
3. Use a text editor to edit the libvirt_volume_drivers and volume_drivers key in the Nova configuration file /etc/nova/nova.conf. (see User Guide for details).
4. Copy the file scaleio.filters to the location /etc/nova/rootwrap.d (create folder if not exist)

CINDER

1. Install and configure the SDC component.
SDC is required on the Cinder host for the operations: copy volume to image, and copy image to volume.
2. Copy the file scaleio.py to /cinder/volume/drivers/emc
3. Copy the file scaleio.filters to the location /etc/cinder/rootwrap.d
4. Use a text editor to edit lines in the file cinder.config in /etc/cinder/cinder.conf (see User Guide for details)
5. Create a ScaleIO driver configuration file called /etc/cinder/cinder_scaleio.config, containing parameters for the REST Gateway, Protection Domain and Storage Pool.

For example:

```
[scaleio]
rest_server_ip=10.76.xx.xx
rest_server_username=admin
rest_server_password=<password>
protection_domain_name=pd1
storage_pool_name=default
storage_pools=pd1:default
round_volume_capacity=False
force_delete=True
```

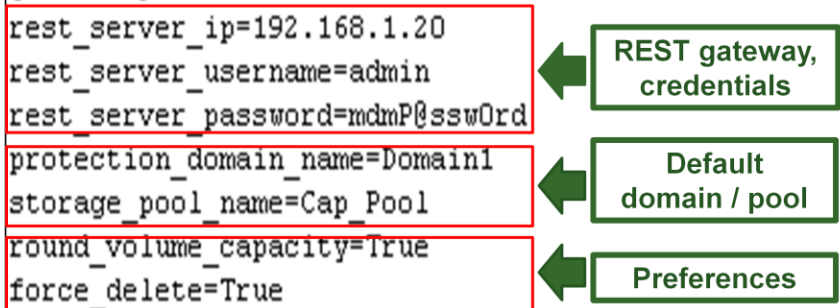
6. Restart the cinder-volume and nova-compute services, using the commands:

```
service openstack-nova-compute restart
service openstack-cinder-volume restart
```

Default (Static) Configuration - `scaleio_cinder.conf`

`/etc/cinder/scaleio_cinder.conf` supplies information OpenStack needs to provision from ScaleIO cluster

```
[root@havana sio(keystone_demo)]# cat cinder_scaleio.config
[scaleio]
rest_server_ip=192.168.1.20
rest_server_username=admin
rest_server_password=mdmP@sswOrd
protection_domain_name=Domain1
storage_pool_name=Cap_Pool
round_volume_capacity=True
force_delete=True
[root@havana sio(keystone_demo)]#
```



EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

144

The `scaleio_cinder.conf` file supplies all information specific to the ScaleIO cluster that is being integrated.

The REST server IP, and the username/password for the ScaleIO cluster enable OpenStack to identify and authenticate to the REST gateway.

By default, OpenStack-created volumes are provisioned from the Protection Domain and Storage Pool specified in the `scaleio_cinder.conf` file.

Other user preference settings can be specified in this file.

Any changes to this file require a full restart of the Cinder Driver in order to take effect.

Identifying Cinder-created Volumes within ScaleIO

- Mapping Cinder-created volume to corresponding volume as reported by ScaleIO:
 - ▶ Perform base-16 to base-64 conversion on Cinder volume ID
 - ▶ **scaleio_config.py -convert_id** may be used for computation
 - ▶ Result is ScaleIO-reported name for volume

```
[root@havana sio(keystone_demo)]# cinder list
```

	ID	Status	Display Name	Size	Volume
e Type	Bootable	Attached to			

	659a49aa-ebc1-4ec7-b809-9eae62590f7f	available	test_vol1	1	N
one	false				

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

145

ScaleIO by itself cannot directly report the Cinder-supplied unique ID to each volume that it creates. However, it is possible to identify the specific ScaleIO volume (as reported by **scli -query_all_volumes** for example) by performing a base-16 to base-64 conversion on the Cinder volume ID that generates the ScaleIO-reported volume name.

The conversion may be conveniently done using the **scaleio_config.py** script with the **-convert_id** option.

Dynamic Configuration - Multiple ScaleIO Storage Tiers

Scaleio_config.py script can be used to map every ScaleIO
Domain/Pool pair to a Cinder **volume type**

```
[root@havana sio(keystone_admin)]# python scaleio_config.py --config --username=
admin --password=1288ab47e76e480c
Options for protection domain and storage pool pairs
*****
number | protection domain | storage pool
-----|-----|-----
1)      | Domain1           | Perf_Pool
2)      | Domain1           | Cap_Pool
3)      | Domain1           | default

Which pair of domain and pool would like to create? (enter a number) 1
Select the name for the new volume type (if left blank, the default of <domain>-
<pool> will be used) Perf_Tier
Volume type name is Perf_Tier
Select the provisioning type = thin or thick (if left blank, the default of thic
k will be used) thin
provisioning type is thin
Select the iops limit in case you want to limit the number of iops one compute n
ode generates for volumes of this volume type (if left blank, no iops limit will
be set)
iops_limit is
Select the bandwidth limit in case you want to limit the network bandwidth that
one compute node generates for volumes of this volume type (if left blank, no ba
ndwidth limit will be set)
bandwidth_limit is
created volume type <VolumeType: Perf_Tier>
extra specs created: (u'sio:pd_name': u'Domain1', u'sio:provisioning': u'thin',
u'sio:sp_name': u'Perf_Pool')
[root@havana sio(keystone_admin)]#
```

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

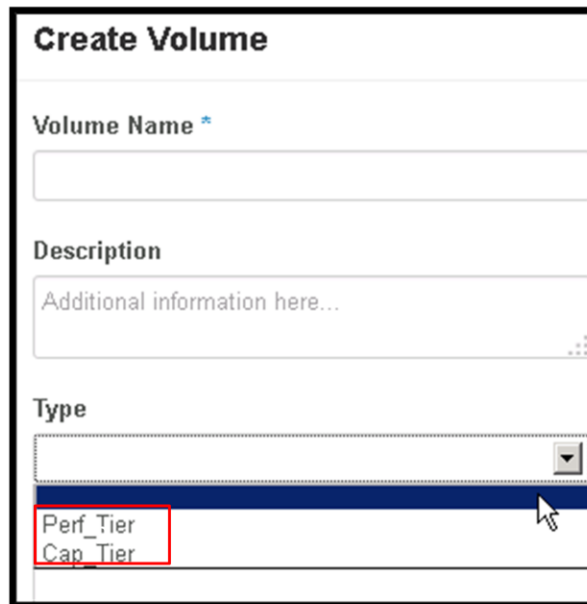
ScaleIO Implementation and Management

146

What we saw earlier with the **scaleio_config.py** was a static configuration, where every volume provisioned from ScaleIO will get provisioned from a specific Protection Domain/Storage Pool pair without having to specify any volume type. This can meet the need of some data centers where all of ScaleIO is viewed as a single storage tier. However, it may be necessary in other environments to specify multiple tiers within ScaleIO.

A simple way to set up a multi-tiered configuration is to create one Cinder volume type for each domain/pool pair. The ScaleIO-provided script **scaleio_config.py** may be used to set up this configuration with minimal effort, as shown. Users must run the script once for each domain/pool pair, and specify a unique cinder type name for each. This gives a dynamic configuration with OpenStack recognizing different tiers of storage within ScaleIO, and identifying them by specific names.

Tiered Storage Configuration - OpenStack View



The screenshot shows the 'Create Volume' form in the OpenStack Horizon UI. It includes fields for 'Volume Name *', 'Description' (with placeholder text 'Additional information here...'), and 'Type'. The 'Type' dropdown menu is open, showing two options: 'Perf_Tier' and 'Cap_Tier'. A red rectangle highlights these two options, and a mouse cursor is pointing at 'Cap_Tier'.

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

147

With a dynamic configuration in place, the volume provisioning interface of the OpenStack Horizon UI will display the Cinder volume types assigned previously, as shown in this example. At the time of volume creation, the user may select any of these types for the particular volume being created.

Dynamic Configuration - Configurable Volume Attributes

Attribute Name	Description
sio:pd_name	Name of the Protection Domain
sio:sp_name	Name of the Storage Pool
sio:provisioning_type	thick or thin provisioning (default: thick)
sio:iops_limit	Max IO/sec for volume: specify an integer >10 (default: 0 => unlimited)
sio:bandwidth_limit	Max MB/sec for volume: specify an integer > 0 (default: 0 =>unlimited)



Beyond basic tiering by domain/Storage Pool, it is possible to create custom tiers with multiple variations of all the attributes shown in the table, including thin versus thick, IOPs limit and/or bandwidth limit. To do this, the native Cinder commands **type-create** and **type-key** must be used. Refer to the ScaleIO User Guide for details.

ScaleIO with OpenStack - Deployment Considerations

- Supported OpenStack releases: Icehouse, Juno
- KVM hypervisor only
 - ▶ No support for VMware/Hyper-V as hypervisors
- SDC client required on all Nova nodes and cloud controller node
- Live migration of VMs supported
- Requires SSL access from cloud controller to ScaleIO gateway
 - ▶ Default: Cinder Driver does not verify SSL certificate of gateway
 - ▶ Can be configured to perform verification



Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

149

Some of the key deployment considerations for ScaleIO integration with OpenStack are listed here. Note that both the Havana and the Icehouse releases are supported with ScaleIO v1.30. The Nova nodes and the cloud controller node need to run a supported Linux distribution, since they require SDC client to be installed on them. Provisioning to VMs can only be done for VMs on the KVM hypervisor when VMware or Hyper-V are configured as alternate hypervisors within Nova. VMs on those hypervisors cannot be supported. For KVM-based VMs, live migration with ScaleIO providing the shared storage is supported.

By default, the ScaleIO Cinder Driver does not verify the SSL certificate of the REST gateway when it attempts to connect to it via SSL. This can be reconfigured to enforce verification. Please refer to the Scale IO v1.30 User Guide for details.

This concludes our discussion of the the ScaleIO REST gateway, and integration with OpenStack.

Lab Exercises

Lab 5b: ScaleIO 1.30 integration with OpenStack

Refer to the *Student Lab Guide*

Purpose:

Integrate a ScaleIO cluster with an OpenStack cloud environment, enabling ScaleIO provisioning from native OpenStack user interfaces.

EMC²

Copyright © 2015 EMC Corporation. All Rights Reserved.

ScaleIO Implementation and Management

150

Please proceed to Lab Exercise 5b, ScaleIO Integration with OpenStack.