Storage Networking Industry Association
Technical White Paper

Spare Resource and Capacity Overview

August 2018

Abstract: *The increased sophistication of storage systems has led to a similar evolution of data redundancy and capacity management. This paper distinguishes hot spare disks from hot spare capacity. The use of a RecoverableCapacitySourceCount is highlighted as a more flexible and user-focused assessment of system resiliency.*

USAGE

The SNIA hereby grants permission for individuals to use this document for personal use only, and for corporations and other business entities to use this document for internal use only (including internal copying, distribution, and display) provided that:

1. Any text, diagram, chart, table or definition reproduced shall be reproduced in its entirety with no alteration, and,

2. Any document, printed or electronic, in which material from this document (or any portion hereof) is reproduced shall acknowledge the SNIA copyright on that material, and shall credit the SNIA for granting permission for its reuse.

Other than as explicitly provided above, you may not make any commercial use of this document, sell any or this entire document, or distribute this document to third parties. All rights not explicitly granted are expressly reserved to SNIA.

Permission to use this document for purposes other than those enumerated above may be requested by e-mailing tcmd@snia.org. Please include the identity of the requesting individual and/or company and a brief description of the purpose, nature, and scope of the requested use.

All code fragments, scripts, data tables, and sample code in this SNIA document are made available under the following license:

ACKNOWLEDGEMENTS

The SNIA Scalable Storage Management Technical Work Group, which developed and reviewed this document, would like to recognize the significant contribution made by the principal author, Doug Voigt.

Member	Representative
Broadcom Limited	Richelle Ahlvers
Dell Inc.	Patrick Boyd George Ericson Michael Raineri Rich Roscoe
Hitachi Data Systems	Eric Hibbard
Hewlett Packard Enterprise	Jeff Hilland Chris Lionetti John Mendonca Doug Voigt
Inova Development Inc.	Karl Schopmeyer
Intel Corporation	Slawek Putyrski Paul von Behren
Microsemi	Anand Nagarjan
Microsoft Corporation	Hector Linares Jim Pinkerton Michael Pizzo Scott Seligman
NetApp, Inc.	Don Deel Nilesh Maheshwari
ScienceLogic	Patrick Strick
VMware, Inc.	Murali Rajagopal

DISCLAIMER

The information contained in this publication is subject to change without notice. The SNIA makes no warranty of any kind with regard to this specification, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The SNIA shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this specification.

Suggestions for revisions should be directed to http://www.snia.org/feedback/.

Executive Summary

The increased sophistication of storage systems has led to a similar evolution of data redundancy and capacity management. This paper distinguishes hot spare disks from hot spare capacity. The use of a RecoverableCapacitySourceCount is highlighted as a more flexible and user-focused assessment of system resiliency.

Evolution of Spares and RAID

In any RAID system, redundancy can be used to recover data on a failed disk drive (HDD or SSD). In addition to satisfying host requests in degraded mode, simple RAID implementations can use a spare disk (provided that one is available), and rebuild the image of the data from the failed disk. When the rebuild completes, a new spare drive is needed in order to restore redundancy, and be able to repeat the process after any future failure.

More recently many RAID implementations have gotten smarter. They distribute spare capacity throughout all of the disks attached to the array so there is no dedicated spare drive. The advantage is better performance because all disks are being used rather than leaving the spare drive unused. Since both types of implementations still exist there is a dichotomy between spare drives and spare capacity.

The purpose of this document is to clarify and augment the treatment of spare storage in Redfish and Swordfish. To do this we will:

- Describe the distinction between spare resources such as drives and spare capacity in a system.
- Describe how resources such as volumes, storage pools, and file systems can be associated with shared sets of spare components.
- Describe how clients can specify a desired recoverable capacity source count as a data storage line of service capability, and determine its maximum and current values.

We define “active spare capacity” to describe implementations that distribute spare capacity, intermingling it on the same drives as user data.

> Active Spare Capacity – Capacity in a resource such as a StoragePool that can be used to complete a rebuild of a failing or inaccessible drive. This capacity can be distributed across disk drives or memory components, none of which is a HotSpare.

Active spare capacity is separate from hot spare drives. Until a failover occurs, a hot spare drive is inactive and cannot be used for user data. Active spare capacity, on the other hand, is tracked at a finer grain than whole disk drives, so that a drive can satisfy user-level IO requests while preserving the capacity within that drive that is needed for failure recovery.

Spare Drives

Focusing first on spare drives, there are two approaches: the dedicated spare drive and the spare resource set approach. Now we will describe each one.

Dedicated Spares

Dedicated spare drives in Redfish are based on the definition of HotSpare.

> HotSpare – unused storage resource (e.g. drive) that can be called into use without manual intervention. There is no user data on a HotSpare drive.

HotSpare drive modes are described in the Drive.HotSpareType enum as follows:

- None - The drive is not currently a HotSpare.
- Global - The drive is currently serving as a HotSpare for all other drives in the storage system.
- Chassis - The drive is currently serving as a HotSpare for all other drives in the chassis.
- Dedicated - The drive is currently serving as a HotSpare for a user defined set of drives.

When the Drive.HotSpareType equals Dedicated, the user defined set of drives is the Volume.Links.DedicatedSpareDrives collection in a volume which is defined as follows:

> **Volume.Links.DedicatedSpareDrives** -- a reference to the Drive resources that that are currently assigned as a dedicated spare and are able to support this Volume.

The Volume.Links.DedicatedSpareDrive collection is designated with read permission only. Clients should not post drives to this collection as the scope of the spare drive (i.e. the set of volumes for which it can act as a spare) is implementation specific.

Spare Resource Sets

As of Swordfish v.1.0.7

The purpose of a spare resource set is to locate spare resources and their consumers. Spare resources may be on line or off line; they may have different types and reside in multiple places.

When collections of spare resource sets are referenced by volumes, storage pools or file systems, the spare resource sets must have resource types that are potential sources of capacity for the referencing resources.

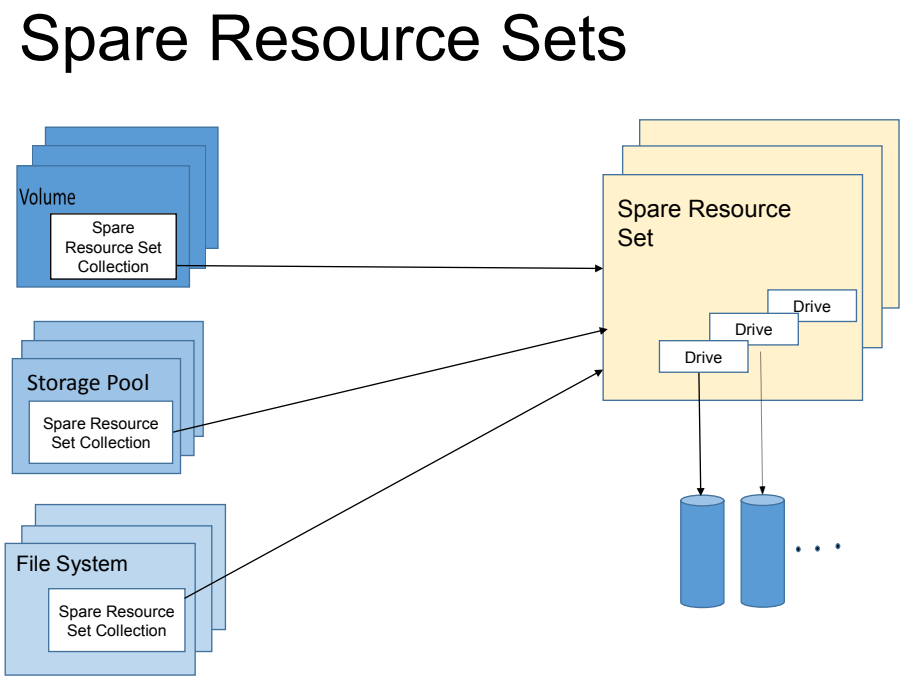

Figure 1: Spare Resource Sets

In Figure 1, we see that a spare resource set is linked to a set of drives. Collections of spare resource sets can be contained in volumes, storage pools and file systems. This allows a many-to-many relationship between spare resource sets and the potential consumers of spare drives. The current definition of a SpareResourceSet is summarized in Table 1.

Table 1 SpareResourceSet Properties

SpareResourceSet		
	ResourceType	The type of resources in the set. (e.g. strings referring to drives, memory, volumes, controllers, fans, power supplies, etc.)
	TimeToProvision	Amount of time needed to make an on-hand resource available as a spare.
	TimeToReplenish	Amount of time to needed replenish consumed on-hand resources.
	OnHandLocation	The location where this set of spares is kept. (from Resource.Location)
	OnLine	This set shall be online.
	Links	This structure shall contain references to resources that are not contained within this resource.
Links		
	OnHandSpares	The type of resources in the set.
	ReplacementSpareSets	Other spare sets that can be utilized to replenish this spare set.

RecoverableCapacitySourceCount Motivation and Definition

There are many good reasons why the industry has moved to active spare capacity rather than spare drives. Today's arrays distribute spare capacity in so many different ways that it is difficult to track from outside of the array. In all cases, however, the ultimate goal is to be able to survive failures and complete rebuilds.

For implementations that use active spare capacity, the RecoverableCapacitySourceCount capability (RCSC) enables clients to assure that rebuilds could complete in the event of a specified number of drive failures. The use of an abstract count is important because active spare capacity and hot spare drives are separate, and both can be used to recover from failures. Given possible permutations of available active spare capacity, hot spare drives and redundancy algorithms, there was no common way to ascertain

assurance of recovery including rebuild completion. RecoverableCapacitySourceCount answers that need.

> **RecoverableCapacitySourceCount** - the number of times, given the resources available to the system, that a desired level of redundancy can be restored (e.g. via rebuild) after a failure.

Within this definition, capacity sources include disk drives, SSD's, and persistent memory components. "Available" means discoverable, accessible, and usable, without manual intervention. This is intended to be a conservative metric. Accordingly, the count shall be calculated based on the most significant potential failure among the contributing sources of capacity, such as the largest disk drive in a RAID set. Further, the RCSC does not account for rates of failure that exceed the rate of rebuild. Therefore, no attempt is made to account for failures that occur too rapidly to allow sufficient rebuilding to complete before data loss.

RecoverableCapacitySourceCount appears in several locations of the Swordfish schema:

- MaximumRecoverableCapacitySourceCount appears in the DataStorageLoSCapabilities resource to indicate the maximum value of that capability supported by an implementation.
- DesiredRecoverableCapacitySourceCount appears in the DataStorageLineOfService resource in ClassOfService to indicate the expectation of that capability when that class of service is in affect
- CurrentRecoverableCapacitySourceCount appears in the Volume, FileSystem and StoragePool resources to indicate the actual value of that capability that can be fulfilled for that resource given the current system state. It is assumed that drives and memory components can be replaced, repaired or otherwise added to increase the CurrentRecoverableCapacitySourceCount.

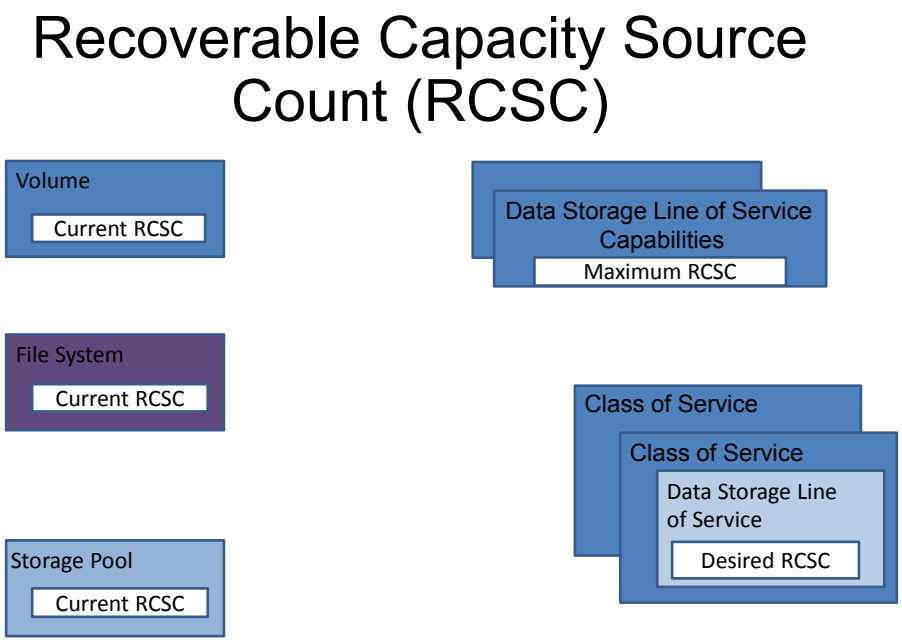

Figure 2: RCSC

Figure 2 shows the places where RecoverableCapacitySourceCounts (RCSC) can appear in Swordfish. Classes of service indicate administrative goals in the form of desired RCSC, while capabilities indicate the maximum value that the system is designed to support. Volumes, file systems, and storage pools each contain indications of a current RCSC based on the state of the part of the system that is relevant to each. Note that the desired RCSC is always between 0 and the maximum. The current RCSC has the same potential range of values, but it varies in real time because it is tied to available capacity by the array implementation.

Figure 3 illustrates some of the interplay between the current RCSC and the desired RCSC. The RCSC lifecycle can include a broad range of conditions. It is even possible for the current RCSC to be larger than the desired level, if excess spare capacity is available. At the same time, the RCSC lifecycle is intended to model overall system resources, and does not reflect device-specific behaviors, such as SSD wear leveling, or other device management algorithms.

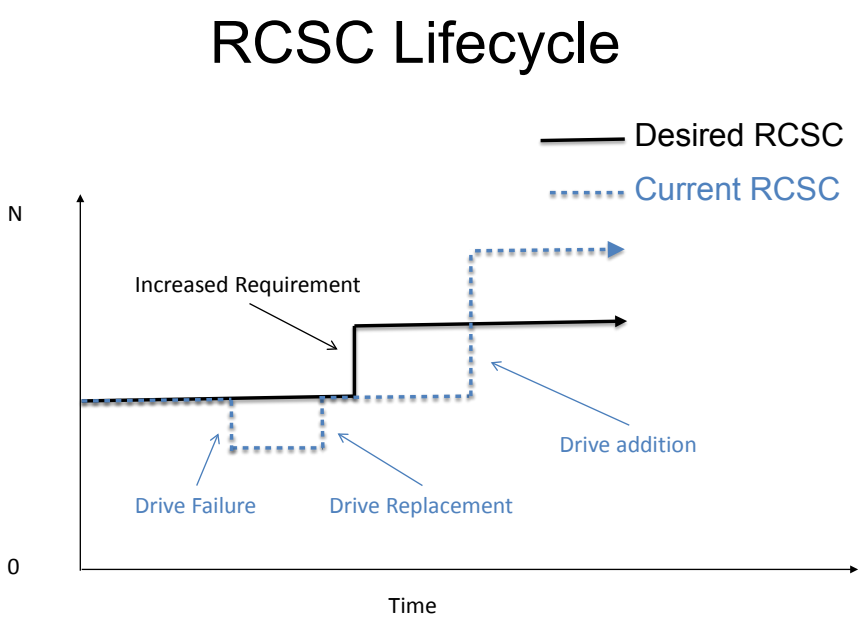

Figure 3: RCSC Lifecycle

Treatment of Memory Resources

Memory resources are allowed to appear in spare resource sets. Since they are sources of capacity they can appear in collections of spare resource sets referenced by volumes, file systems and storage pools.

The RecoverableCapacitySourceCount capability includes recovery from failures that affect memory capacity, when it is delivered as capacity sources with automated redundancy and rebuild implementations.

Summary

With dedicated hot spare drives, spare resource sets and Recoverable Capacity Source Counts, customers have the tools to manage spare resources as they see fit. Dedicated spares are good for very simple configurations. Spare resource sets enable the flexibility to share spare resources. Recoverable Capacity Source Counts allow arbitrarily complex automated spare capacity management to be governed from a Class of Service standpoint.

For More Information

More information about Swordfish, and the SNIA Scalable Storage Management API can be found at http://www.snia.org/swordfish.