

# EMC Data Domain Data Invulnerability Architecture: Enhancing Data Integrity and Recoverability

*A Detailed Review*

---

## **Abstract**

No single mechanism is sufficient to ensure data integrity in a storage system. It is only through the cooperation of a multitude of mechanisms that establish successive lines of defense against all sources of errors that data recoverability can be assured. Unlike traditional general-purpose storage systems, EMC<sup>®</sup> Data Domain<sup>®</sup> systems have been designed explicitly for data protection. This white paper focuses on the four key elements of the Data Domain Data Invulnerability Architecture, which, in combination, provide the industry's highest levels of data integrity and recoverability:

- End-to-end verification
- Fault avoidance and containment
- Continuous fault detection and healing
- File system recoverability

August 2010

---

---

Copyright © 2010 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.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on [EMC.com](http://EMC.com)

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

Part Number h7219.1

---

## Table of Contents

<b>Executive summary .....</b>	<b>4</b>
Storage system data integrity .....	4
<b>Introduction .....</b>	<b>4</b>
Audience .....	4
<b>Data Domain Data Invulnerability Architecture .....</b>	<b>4</b>
End-to-end verification .....	5
Fault avoidance and containment .....	5
New data never overwrites good data .....	5
Fewer complex data structures .....	5
NVRAM for fast, safe restart .....	6
No partial stripe writes .....	6
Continuous fault detection and healing .....	6
RAID 6: Double disk failure protection .....	7
On-the-fly error detection and correction .....	7
Scrub to ensure data doesn't go bad .....	7
File system recoverability .....	7
Self-describing data format to ensure metadata recoverability .....	8
FS check, if needed, is fast .....	8
<b>Conclusion .....</b>	<b>8</b>

---

## Executive summary

### ***Storage system data integrity***

Behind all their added value, specialized storage systems are built on software and general-purpose computing components that can all fail. Some failures have an immediate visible impact, such as the total failure of a disk drive. Other failures are subtle and hidden, such as a software bug that causes latent file system corruption that is only discovered at read time. To ensure data integrity in the face of such failures, the best storage systems include various data integrity checks and are generally optimized for performance and system availability, not data invulnerability. In the final analysis, they assume that backups get done, and make design tradeoffs that favor speed over guaranteed data recoverability. For example, no widely used primary storage file system reads data back from disk to ensure it was stored correctly; to do so would compromise performance. But data can't be considered invulnerable if it isn't stored correctly in the first place. In backup-to-disk, the priority must be data invulnerability over performance and even availability. Unless the focus is on data integrity, the backup data is at risk. If the backup data is at risk, then when the primary copy of the data is lost, recovery is at risk. Most backup-to-disk storage systems are just primary storage systems built out of cheaper disks. As such, they inherit the design philosophy of their primary storage predecessors. Though labeled as backup-to-disk products, their designs emphasize performance at the expense of data invulnerability.

## Introduction

This white paper focuses on the four key elements of the EMC® Data Domain® Data Invulnerability Architecture, which, in combination, provide the industry's highest levels of data integrity and recoverability.

### ***Audience***

This white paper is intended for EMC customers, technical consultants, partners, and members of the EMC and partner professional services community who are interested in learning more about the Data Domain Data Invulnerability Architecture.

## Data Domain Data Invulnerability Architecture

Data Domain deduplication storage systems represent a clean break from conventional storage system design thinking and introduce a radical premise: What if data integrity and recoverability was the most important goal? If one imagines a tapeless IT department, one would have to imagine extremely resilient and protective disk storage. Data Domain systems have been designed from the ground up to be the storage of last resort. The Data Domain operating system (DD OS) is purpose-built for data invulnerability. There are four critical areas of focus:

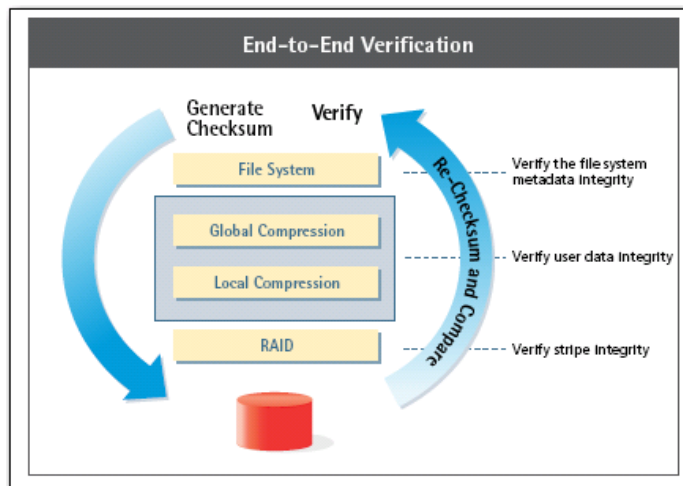
- End-to-end verification
- Fault avoidance and containment
- Continuous fault detection and healing
- File system recoverability

Even with this model, it is important to remember that DD OS is only as good as the data it receives. It can do an end-to-end test of the data it receives within its system boundaries, but it cannot know whether that data has been protected by all steps in the network on the way to it. If there is an error in the backup network that causes data corruption, or if the data is corrupted in place in primary storage, DD OS cannot repair it. It remains prudent to test recovery to the application level on a periodic basis.

## End-to-end verification

Since every component of a storage system can introduce errors, an end-to-end test is the simplest path to ensure data integrity. End-to-end verification means reading data after it is written and comparing it to what was sent to disk, proving that it is reachable through the file system to disk, and proving the data is not corrupted. When DD OS receives a write request from backup software, it computes a huge checksum over the constituent data. After analyzing the data for redundancy, it stores the new data segments and all of the checksums. After the I/O dust has settled on a backup and all the data has been synched to disk, DD OS verifies that it can read the entire file from the disk platter and through the Data Domain file system, and that the checksums of the data read back match the checksums of the written data. This ensures that the data on the disks is readable and correct and that the file system metadata structures used to find the data are also readable and correct. This confirms the data is correct and recoverable from every level of the system. If there are problems

anywhere along the way, for example if a bit has flipped on a disk drive, it will be caught. For the most part, it can be corrected through self-healing as described below in the next section. If for any reason it can't be corrected, it will be reported immediately, and a backup can be repeated while the data is still valid on the primary store. Conventional, performance-optimized storage systems cannot afford such rigorous verifications. Backup-to-disk requires them. The tremendous data reduction achieved by Data Domain Global Compression™ reduces the amount of data that needs to be verified and makes such verifications possible.



**Figure 1. The end-to-end check verifies all file system data and metadata.**

## Fault avoidance and containment

The next step in protecting the data is to make sure the data which was verified to be correct stays correct. Ironically, the biggest risk to file system integrity is file system software errors when writing new data. It is only new writes that can accidentally write on existing data, and new updates to file system metadata that can mangle existing structures. Because the Data Domain file system was built to protect data as its primary goal, its design protects even against its own software errors that could put existing backups at risk. It accomplishes this through a combination of design simplicity—which reduces the chance of bugs in the first place—and several fault containment features, which make it difficult for potential software errors to corrupt existing data. Data Domain systems are equipped with a specialized log-structured file system that has four important benefits.

### New data never overwrites good data.

Unlike a traditional file system, which will often overwrite blocks when data changes, Data Domain systems only write to new blocks. This isolates any incorrect overwrite (a software bug type of problem) to only the newest backup data. Older versions remain safe.

### Fewer complex data structures

In a traditional file system, there are many data structures (e.g., free block bit maps and reference counts) that support very fast block updates. In a backup application, the workload is primarily simpler sequential writes of new data, meaning fewer data structures are required to support it. As long as the system can keep

---

track of the head of the log, new writes will not touch old data. This design simplicity greatly reduces the chances of software errors that could lead to data corruption.

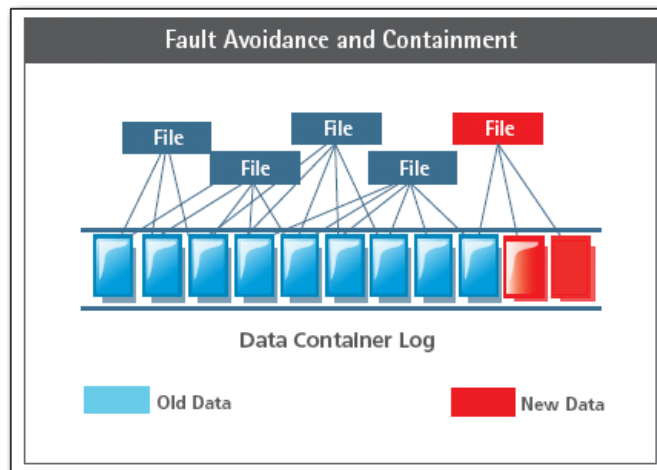
## NVRAM for fast, safe restart

The system includes a non-volatile RAM write buffer into which it puts all data not yet safely on disk. The file system leverages the security of this write buffer to implement a fast, safe restart capability. The file system utilizes many internal logic and data structure integrity checks. If any problem is found by one of these checks, the file system restarts itself. The checks and restarts provide early detection and recovery from data corruption errors. As it restarts, the Data Domain file system verifies the integrity of the data in the NVRAM buffer before applying it to the file system, ensuring that no data is lost due to the restart. Because the NVRAM is a separate hardware device, it protects the data from errors that can corrupt data in RAM. Because the RAM is non-volatile, it also protects against power failures. Though the NVRAM is important for ensuring the success of new backups, the file system guarantees the integrity of old backups even if the NVRAM itself fails.

## No partial stripe writes

Traditional primary storage disk arrays, whether RAID 1, RAID 3, RAID 4, RAID 5, or RAID 6, can lose old data if, during a write, there is a power failure that causes a disk to fail. This is because disk reconstruction depends on all the blocks in a RAID stripe being consistent, but during a block write there is a transition window where the stripe is inconsistent, so reconstruction of the stripe would fail, and the old data on the failed disk would be lost. Enterprise storage systems protect against this with NVRAM or uninterruptible power supplies. But if these fail because of an extended power outage, the old data could be lost and a recovery attempt could fail. For this reason, Data Domain systems never update just one block in a stripe.

Following the no-overwrite policy, all new writes go to new RAID stripes and those new RAID stripes are written in their entirety<sup>1</sup>. The verification after write ensures that the new stripe is consistent. New writes don't put existing backups at risk. Data Domain systems are designed to minimize the number of standard storage system errors. If more challenging faults happen, it takes less time to find them, correct them, and notify the operator.



**Figure 2. New data never puts old data at risk. The data container log never overwrites or updates existing data. New data is always written in new containers (in red). The old containers and references remain in place and are safe even in the face of software bugs or hardware faults that may occur when storing new backups.**

## Continuous fault detection and healing

No matter the software safeguards in place, it is the nature of computing hardware to have occasional faults. Most visibly in a storage system, disk drives can fail. But other more localized or transient faults also occur. An individual disk block may be unreadable or there could be a bit flip on the storage

---

<sup>1</sup> The gateway product, which relies on external RAID, is unable to guarantee that there are no partial stripe writes.

---

interconnect or internal system bus. For this reason, DD OS builds in extra levels of data protection to detect faults and recover from them on-the fly and to ensure successful data restore operations.

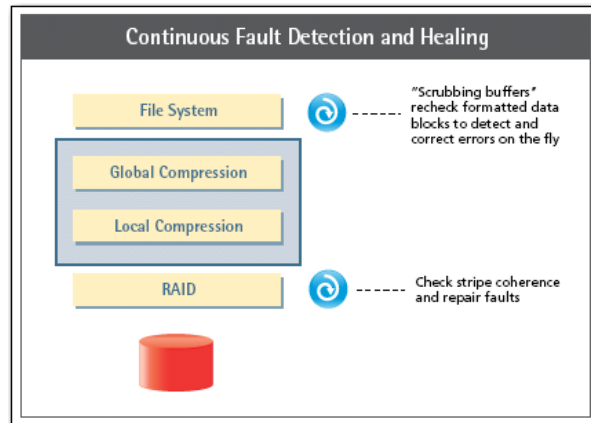
## RAID 6: Double disk failure protection, read error correction

RAID 6 is the foundation for Data Domain's continuous fault detection and healing. Its powerful dual-parity architecture offers significant advantages over conventional architectures including RAID 1 (mirroring), RAID 3, RAID 4, or RAID 5 single-parity approaches.

RAID 6:

- protects against two disk failures
- protects against disk read errors during reconstruction
- protects against the operator pulling the wrong disk
- guarantees RAID stripe consistency even during power failure without reliance on NVRAM or UPS
- verifies data integrity and stripe coherency after writes

By comparison, once a single disk is down in the other RAID approaches, any further simultaneous disk error will cause data loss. A system whose focus is data protection must include the extra level of protection RAID 6 provides.



**Figure 3. Continuous fault detection and healing protect against storage system faults. The system periodically re-checks the integrity of the RAID stripes and the container log and uses the redundancy of the RAID system to heal any faults. During every read, data integrity is re-verified and any errors are healed on the fly.**

## On-the-fly error detection and correction

To ensure that all data returned to the user during a restore is correct, the Data Domain file system stores all of its on-disk data structures in formatted data blocks. These are self-identifying and covered by a strong checksum. On every read from disk, the system first verifies that the block read from disk is the block expected. It then uses the checksum to verify the integrity of the data. If any issue is found, it asks RAID 6 to use its extra level of redundancy to correct the data error. Because the RAID stripes are never partially updated, their consistency is ensured, as is the ability to heal an error when it is discovered.

## Scrub to ensure data doesn't go bad

On-the-fly error detection works well for data that is being read, but it does not address issues with data that may be unread for weeks or months before it is needed for a recovery. For this reason, Data Domain systems actively re-verify the integrity of all data every week in an ongoing background process. This scrub process will find and repair grown defects on the disk before they can become a problem. Through RAID 6, on-the-fly error detection and correction, and ongoing data scrubbing, most computing-system and disk drive-generated faults can be isolated and overcome with no impact on system operation or data risk.

## File system recoverability

Though every effort is made to ensure there are no file system issues, the Data Invulnerability Architecture anticipates that, being man-made, some system some time may have a problem. It therefore includes features to reconstruct lost or corrupted file system metadata and also file system check tools that can bring an ailing system safely back online quickly.

---

## Self-describing data format to ensure metadata recoverability

Metadata structures, such as indices that accelerate access, are rebuildable from the data on disk. All data is stored along with metadata that describes it. If a metadata structure is somehow corrupted, there are two levels of recoverability. First, a snapshot is kept of the file system metadata every several hours; recoverability can rely on this point-in-time copy. Second, the data can be scanned on disk and the metadata structure can be rebuilt. These capabilities enable recoverability even if there is a worst case corruption of the file system or its metadata.

### FS check, if needed, is fast

In a traditional file system, consistency is not checked online at all. Data Domain systems check through initial verification after each backup to ensure consistency for all new writes. The usable size of a traditional file system is often limited by the time it would take to recover the file system in the event of some sort of corruption. Imagine running fsck on a traditional file system with more than 80 TB of data. The reason the checking process can take so long is that the file system needs to sort out where the free blocks are so that new writes don't end up overwriting existing data accidentally. Typically this entails checking all references to rebuild free block maps and reference counts. The more data in the system, the longer this takes. In contrast, since the Data Domain file system never overwrites old data and doesn't have block maps and reference counts to rebuild, it only has to verify where the head of the log is to safely bring the system back online to restore critical data.

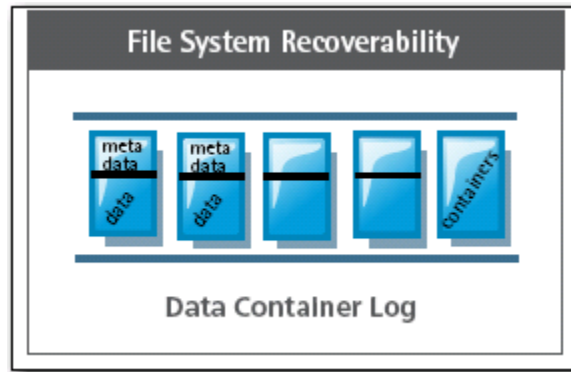
## Conclusion

No single mechanism is sufficient to ensure data integrity in a storage system. It is only through the cooperation of a multitude of mechanisms that establish successive lines of defense against all sources of errors that data recoverability can be assured.

Unlike a traditional storage system that has been repurposed from primary storage to data protection, Data Domain systems have been designed from the ground up explicitly for data protection. The innovative Data Invulnerability Architecture lays out the industry's best defense against data integrity issues. Advanced verification ensures that new backups are stored correctly. The no-overwrite, log-structured architecture of the Data Domain file system together with the insistence on full-stripe writes ensures that old backups are always safe even in the face of software errors during new backups. Meanwhile, a simple and robust implementation reduces the chance of software errors in the first place.

The above mechanisms protect against problems during the storage of backups, but faults in the storage itself also threaten data recoverability. For this reason, the Data Invulnerability Architecture includes a proprietary implementation of RAID 6 that protects against up to two disks failures, can rebuild a failed disk even if there is a data read error, and corrects errors on-the-fly during read. It also includes a continuous scrub process that actively seeks out and repairs latent faults before they become a problem.

The final line of defense is the recoverability features of the Data Domain file system. The self-describing data format enables the reconstruction of file data even if various metadata structures are corrupted or lost. And, the fast file system check and repair means that even a system holding dozens of terabytes of data won't be offline for long if there is some kind of problem.



**Figure 4. Data is written in a self-describing format. If necessary the file system can be re-created by scanning the log and rebuilding it from the metadata stored with the data.**