WHITE PAPER

HIGH AVAILABILITY, DATA PROTECTION, AND DATA INTEGRITY IN THE DELL EMC XTREMIO X2 ARCHITECTURE

Abstract

A key function of an enterprise-class storage system is to host data in a safe and reliable manner. The storage system must provide continuous, uninterrupted access to data, meet stringent performance requirements, and deliver advanced functionality to streamline operations and simplify data management.

This white paper defines the high availability, data protection and data integrity, and examines how XtremIO's unique hardware and software design have been successful in enabling it to exceed 99.999% in uptime and resiliency against failures.

This document also details the monitoring, redundancy levels, integrity checks, and the extreme architectural flexibility to maintain system performance and data availability, by adjusting to failures.

August, 2017

Contents

Executive Summary

A major goal of every enterprise storage system is to host data in a safe and reliable manner. The storage system must provide continuous, uninterrupted access to data, meet stringent performance requirements, and deliver advanced functionality to streamline operations and simplify data management.

An enterprise storage system must provide the upmost resiliency and have no single point of failure, while protecting data during nearly every imaginable failure. The system should also provide high service levels in the face of component failures.

Even specialized storage systems are built on software and general-purpose computing components that can all fail. Some failures may be immediately visible, such as a disk or SSD failure. Others can be subtle, such as not having enough memory resources, resulting in performance issues. To ensure high availability and data integrity in such failures, the best storage systems have an architecture that maintains I/O flow as long as data protection is not at risk, and includes various data integrity checks that are generally optimized for system performance.

This white paper defines the high availability, data protection and data integrity, and examines how XtremIO's unique hardware and software design exceeds 99.999% in uptime and resiliency against failures.

This document also details the monitoring, redundancy levels, integrity checks, and the extreme architectural flexibility to maintain system performance and data availability, by adjusting to failures.

Introduction

XtremIO's system architecture has been designed from the ground up to provide continuous availability. The array does not have any single point of failure, and is built with enterprise-class protection that enables data to survive all but the most catastrophic events.

The combination of a scale-out design, together with service-oriented and modular software architecture, allows XtremIO to operate as a unified system with the ability to adapt independent modules in the event of unexpected hardware failures.

With over 3,000 customers in some of the most demanding enterprise workloads, XtremIO has a proven record of more than 99.999% reliability, which extends to 99.9999% when combined with EMC VPLEX. The combination of a scale-out design with a service oriented and modular software architecture allows multiple XtremIO X-Bricks to operate as a single system with the ability to adapt independent modules in case of unexpected hardware failures.

Audience

This white paper is intended for Dell EMC customers, technical consultants, partners, and members of the Dell EMC and partner professional services community who are interested in learning more about XtremIO's architecture, to achieve High Availability Data Integrity and Data Protection.

High Availability

High Availability is a system design approach which ensures that service is provided continuously, and with the expected level of performance. Users want their applications to be continuously available. Achieving this goal requires a highly-available storage system. A good design for high availability is one that has enough redundancy to prevent any single point of failure from causing data unavailability. It should also provide enough redundancy to protect against multiple concurrent failures. Enterprise storage systems must also ensure that failures, including unlikely ones, do not result in physical data loss or corruption. Thus, enterprise storage systems should have high levels of failure detection. They should be able to either recover a failed component quickly and automatically, or return to a redundant and balanced state, by changing resource allocation. For example, if two array controllers service data and one of them fails, the system needs to be able to reroute I/O via the remaining controller as quickly as possible.

There are two types of redundancy: Active/Passive and Active/Active. Active/Passive redundancy provisions access components that are idle, and is not operational unless the primary component fails. Two examples of Active/Passive redundancy in enterprise storage systems are:

- Active/Passive controller designs (wherein one controller serves I/O and the other controller does not serve I/O, unless the primary controller fails)
- "Hot spare" drives which are designated spare drives (within the system) that are waiting to be used upon the failure of another drive

In general, an Active/Passive design wastes resources and cost by having additional hardware that is rarely used, yet is part of the system. An Active/Active redundancy design maintains activity on all system components, and ideally balances all of them, achieving the highest resource utilization and the least amount of impact upon any component failure. It is highly desirable to have an Active/Active redundancy system.

While a storage system should always maintain availability upon any single failure, the best system designs do not lose data, even when undergoing dual simultaneous failures.

Data Integrity

The primary function of an enterprise storage system is to reliably store user data. When a host reads data, the storage system must provide the correct data stored at the requested location. The accuracy of the data must be validated from the reception of data by the storage system, and through its passage in the system, until the data is written to the back-end storage medium (end-to-end verification).

To verify that the data is correct upon a read, the system needs to create a fingerprint, which is based upon the stored data, and regularly check the fingerprint when reading the data. The fingerprint ensures that the data has not changed, either at rest or in flight. Ideally, the storage system should use independent locations for the data and its fingerprint. Doing this reduces the probability of any single component affecting the data and fingerprint in the same way, which could lead to a false indication that the data is sound while it is not. The worst thing a storage system can do is to provide corrupted data to the host while indicating that the data is good.

XtremIO's Architecture

Hardware Architecture

The XtremIO's main building block is called an X-Brick. An X-Brick is a highly-available, Active/Active building block with two independent fault-tolerant Storage Controllers (nodes).

X-Bricks can be clustered together to create a large scale-out system that linearly grows in IOPS and bandwidth performance and capacity as more X-Bricks are added. X-Bricks can also scale-up capacity by adding more SSDs to the existing X-Bricks. An X-Brick does not have any single point of failure, thus enabling XtremIO clusters to be established initially, by using just a single X-Brick.

Each X-Brick contains two independent Storage Controllers and a 72-SSD array enclosure. The array enclosure has two Serial Attached SCSI (SAS) DAE Controllers with dual connections to each Storage Controller.

Each XtremIO cluster has two NVRAM components to help vault unwritten data to permanent storage in the event of power failures.

The entire XtremIO array is built using standard components (for example: x86 servers, standard form factor SSDs, and off-the-shelf interface cards) with no proprietary hardware of any kind. This combination of components enables XtremIO to leverage best-of-breed high-quality suppliers, and to benefit from general advances made by the component suppliers.

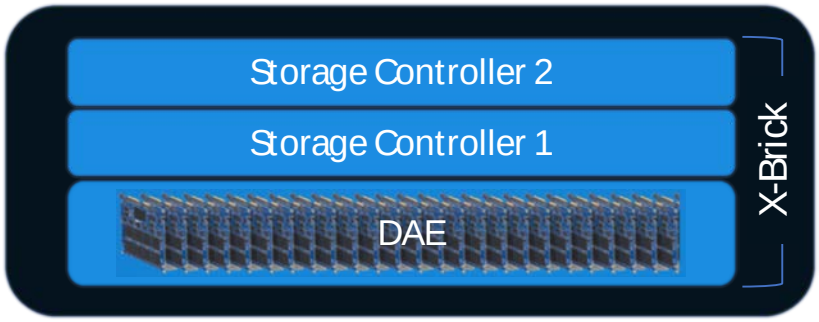


Figure 1. Single X-Brick XtremIO Cluster Hardware

Each XtremIO Storage Controller and disk array enclosure (DAE) is equipped with dual power supplies, each powered from two separate electrical power circuits (in accordance with XtremIO installation best practices).

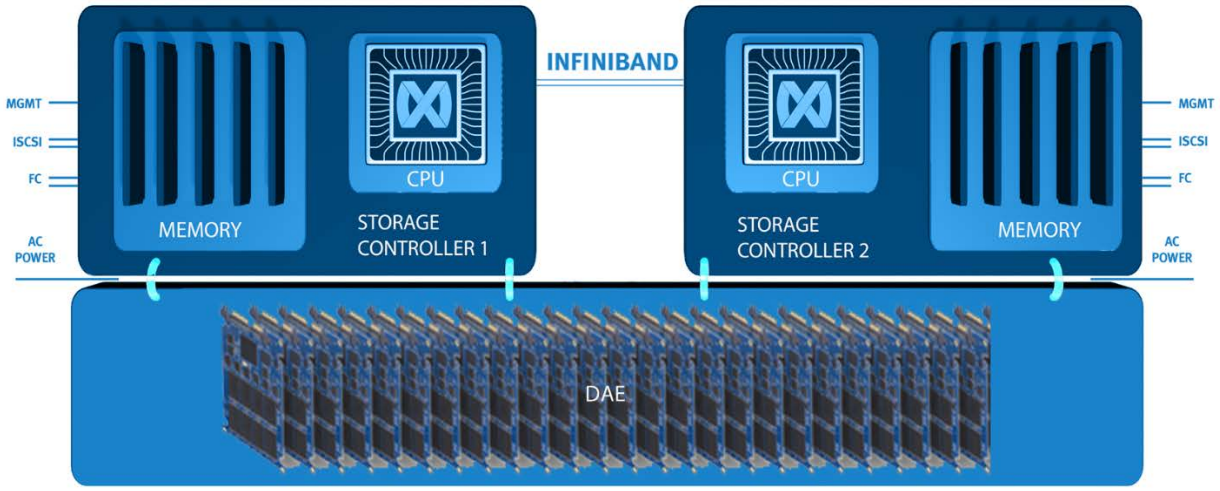


Figure 2. Single XtremIO X-Brick Hardware Logical Block Diagram (use X2 diagram)

A cluster of more than one X-Brick comes equipped with dual InfiniBand Switches. Dual InfiniBand Switches are required for redundancy, and are therefore connected to two separate electrical power circuits. Each Storage Controller is connected to both InfiniBand Switches. The InfiniBand Switches are also connected to each other to provide increased bandwidth and redundancy.

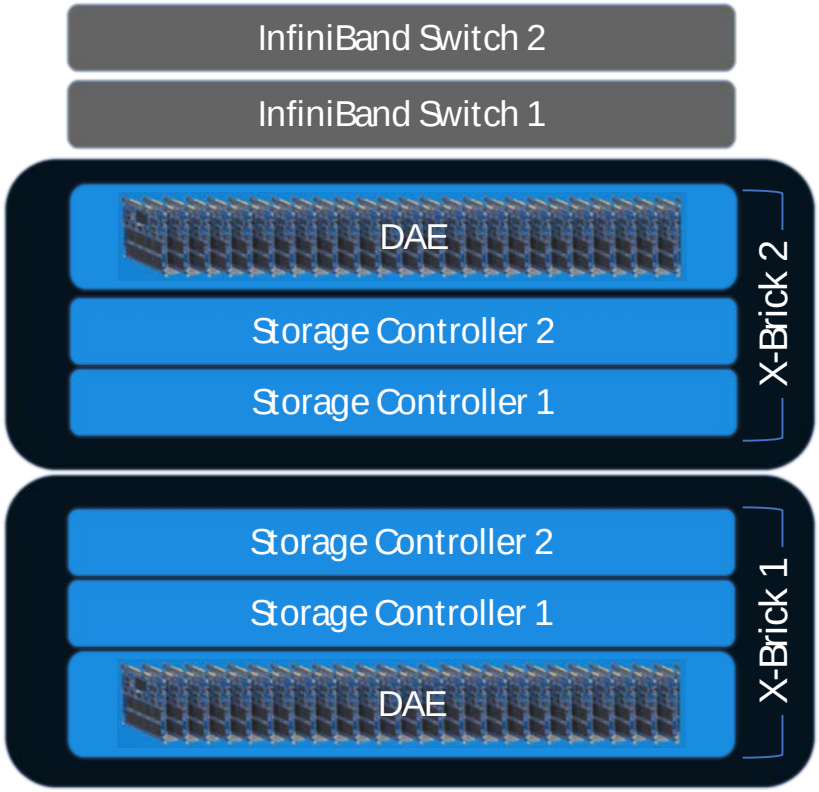


Figure 3. Dual X-Brick XtremIO Hardware

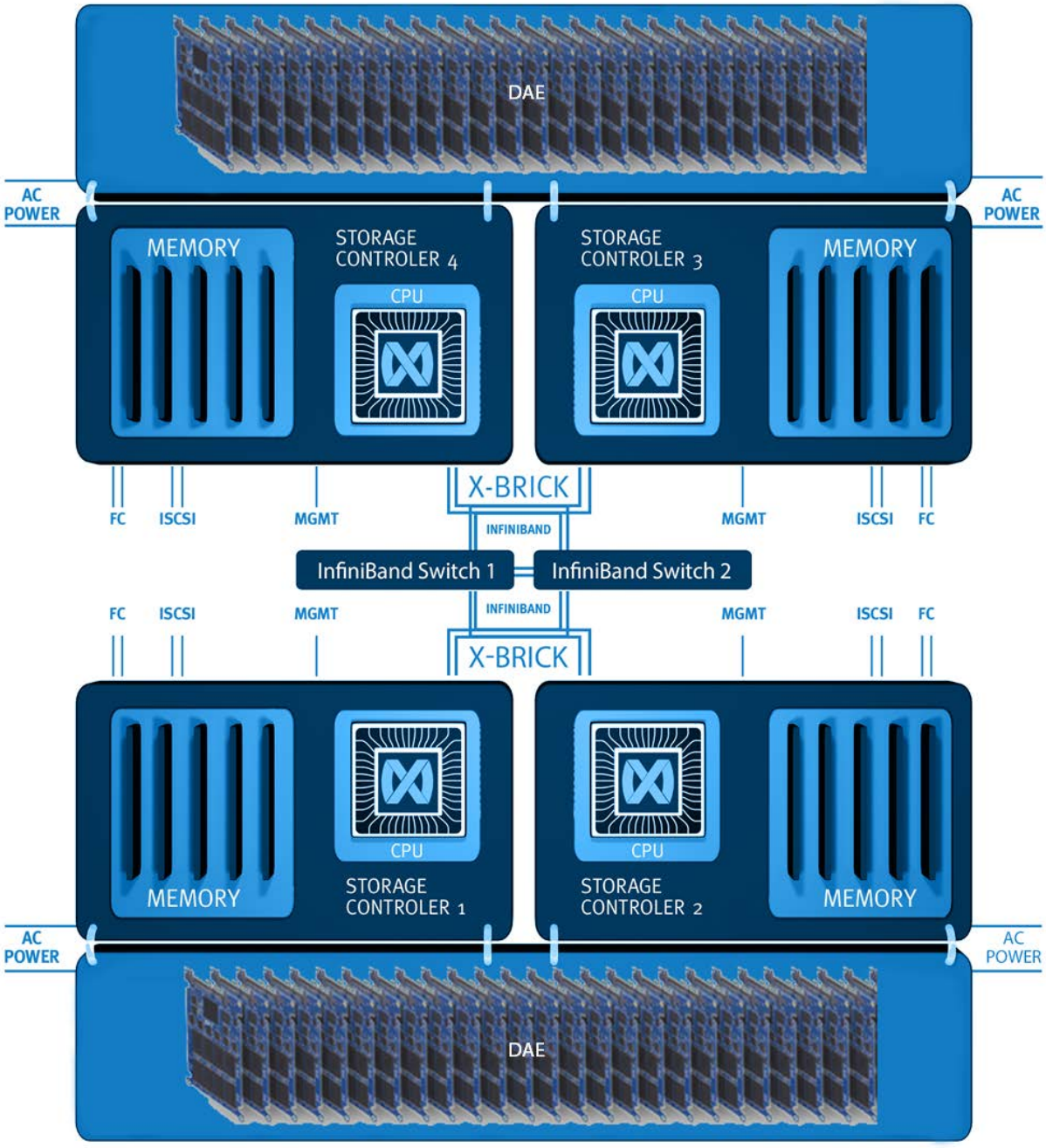


Figure 4. Dual X-Brick XtremIO Hardware Logical Block Diagram

Software Architecture

XtremIO's software architecture enables any software component to run on any Storage Controller in the cluster. This capability allows for continuous operations, despite the occurrence of any hardware failure. The software components within XtremIO are called Modules, and all XtremIO Modules run in user space. Linux provides the underlying kernel. The proprietary operating environment, called XtremIO OS (XIOS), provides scheduling, messaging and special utilities for the XtremIO modules.

There are six main module types within the system, and multiple instances of each one can be running in the system independently. Three module types are infrastructure modules responsible for system-wide management, availability and services for other modules. The other three module types are I/O modules, which are responsible for data services with the array and host communication.

Infrastructure Modules

System-Wide Management (SYM) Module

The System-wide Management module provides a complete view of the hardware and software components. It is responsible for system availability and initiates any changes in system configuration to achieve maximum availability and redundancy.

The SYM module decides as to which modules are to execute on what Storage Controller, initiates failovers of data ownership from one Storage Controller to another, and initiates rebuilds upon SSD failures.

Only one SYM module is the active management entity, and the sole entity that makes system-wide decisions, at any single point in time. Should the component running the active SYM module fail, another SYM module immediately becomes active, and takes over.

An additional software logic runs on each Storage Controller. This additional software is responsible for verifying that one, and only one, SYM is active in the system, a simple process that eliminates the possibility of not having a running SYM module.

Figure 5. XtremIO Storage Controller Software Block Diagram

Platform Manager Module

Each Storage Controller has a single Platform Manager module running. The Platform Manager module is responsible for all activities on the Storage Controller. It monitors the Storage Controller's health, and communicates it to the SYM. The module is responsible for verifying that all processes are running properly in the Storage Controller.

Module shutdowns and restarts are executed by the Platform Manager module on behalf of the SYM module. The Platform Manager module communicates hardware failures to the SYM module. It also facilitates the replicating (journaling) of important data structures between Storage Controllers. It replicates journal memories between Storage Controllers by using Remote Direct Memory Access (RDMA) over the system's InfiniBand fabric. The activity of journaling is critical for redundancy of user data and system metadata.

The Platform Manager module initiates a shutdown of the Storage Controller upon the discovery of loss of power and/or of the complete loss of communication with other Storage Controllers.

I/O Modules

The I/O modules are responsible for storing data from hosts and retrieving the data upon request. An I/O module runs on each Storage Controller. The SYM determines which module runs on which Storage Controller. Every I/O passes through all three types of I/O modules (Routing, Control, and Data).

Routing Module

The Routing module is the system's only entity that communicates with the host. It accepts SCSI commands from the host and parses them.

The Routing module is stateless and simply translates the requests into volumes and Logical Block Addresses (LBAs). The module then forwards a request to the appropriate Control module (and Storage Controller) managing the LBAs.

The Routing module inherently balances the load across the entire XtremIO clustered system, running a content-based fingerprinting function that results in data being evenly distributed across all X-Bricks in the system. For a detailed explanation of this process, refer to the Introduction to the Dell EMC XtremIO Storage Array White Paper.

Control Module

The Control module is responsible for translating the host user address (LBA) to an XtremIO internal mapping. It acts as a virtualization layer between the host SCSI volume/LBA, and the XtremIO back-end deduplicated location. Having this virtualization layer provides an ability to efficiently implement a range of rich data services.

Data stored on XtremIO is content-addressable. The data's array location is determined according to its content. Therefore, it is not based on its address, as is the case with other storage system products. The LBAs of every volume in an XtremIO array are distributed among many Control modules.

Data Module

The Data module is responsible for storing data on the SSDs. It works as a service for the Control module, in which the Control module provides a content fingerprint, and the Data module writes or reads the data according to this fingerprint.

There are only three basic operations that the Data module executes: reading, writing and erasing a block. The goal is to keep the module as simple as possible to maintain a robust and reliable system design. The Control module is not required to deal with XtremIO Data Protection (XDP) allocation. Centralizing the XDP scheme in the Data module provides system-wide flexibility and efficiency.

The Data module evenly maps a content fingerprint to a physical location on an SSD in the same way that the Control module evenly maps a host address to a content fingerprint. This process guarantees that the data is balanced, not only across all Storage Controllers, but also across all SSDs within the array. This additional translation layer also enables the Data module to place the data optimally on the SSDs. Even in challenging scenarios, such as failed components, minimal free space, and frequent data overwrite, XDP can find optimal locations to store data in the system. To learn more about how XDP provides redundancy and flash-optimized data placement, refer to the Dell EMC XtremIO Data Protection White Paper.

Restarting Modules

Since all XtremIO modules run in user space, XIOS can quickly restart modules as needed. Any software failure or questionable module behavior results in an automated module restart. Restarts are non-disruptive, and are generally undetectable at the user level. This capability also serves as the foundation for Non-Disruptive Upgrades (NDUs).

I/O Flow

A Host I/O read request is initially received at the Routing module (R) that parses SCSI. The module then calculates a fingerprint for the data and forwards the I/O request to the relevant Control module (C). The Control module does not have to be physically located in the same Storage Controller as that of the Routing module. The Control module translates the host request into XtremIO's internal data management scheme, and forwards the request to the appropriate Data module (D). The Data module reads the data from the SSDs. The I/O path always follows these exact same steps, regardless of the size of the XtremIO cluster. Thus, the latency remains consistent, regardless of the system scale.

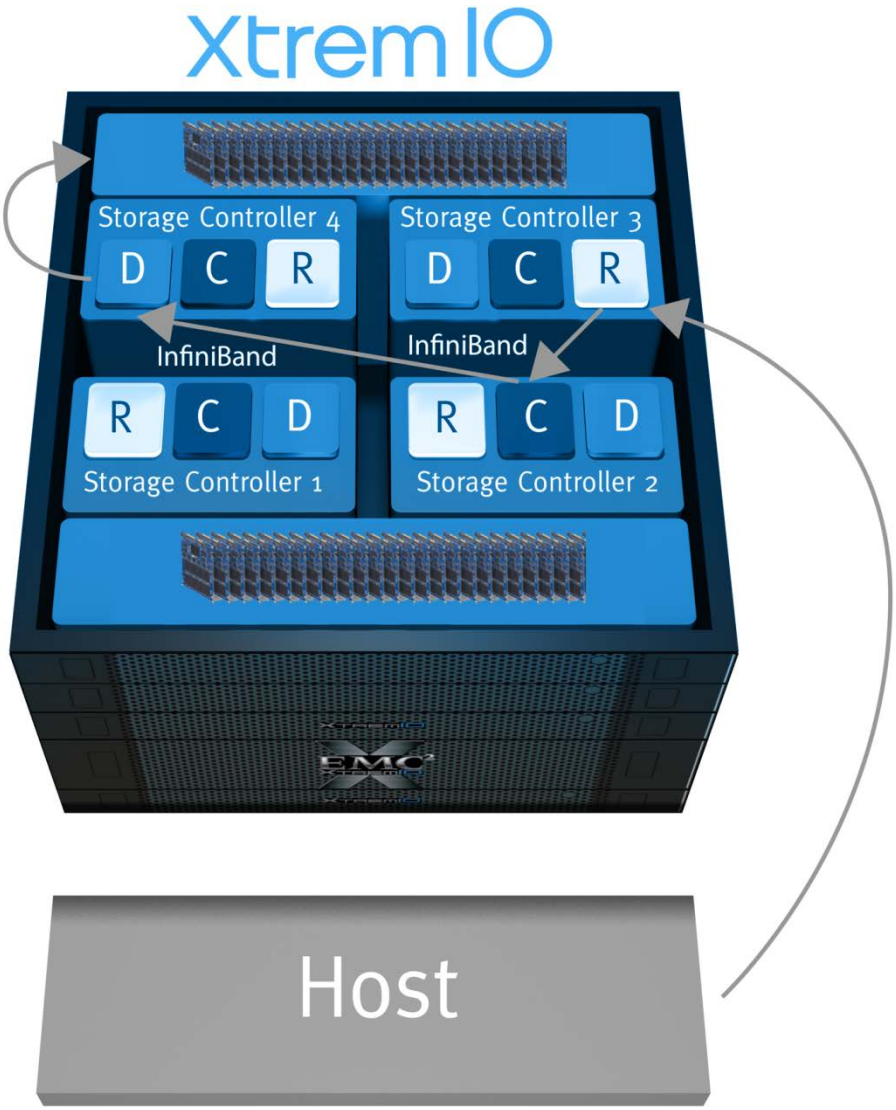


Figure 6. Example of Host Write I/O Flow

Secure Distributed Journaling

As with any enterprise storage system design, the array is not only responsible for protecting data, but also has its own metadata for operation. It is of paramount importance to protect the metadata and maintain coherency. XtremIO has developed a unique distributed journaling mechanism that is designed to protect the entire system's important metadata, and its internal datasets.

A copy of the array's metadata is stored in the Storage Controller memory. Updated metadata is synchronously replicated over InfiniBand remote direct memory access (RDMA) in a distributed fashion to one or more physical Storage Controllers. Therefore, every real-time change is protected in multiple locations.

The System-wide Management module manages the journal's replication relationships between Storage Controllers within the cluster.

For resiliency reasons, a Storage Controller that does not have a healthy NVRAM component cannot be a target for replicated journal data. If a Storage Controller fails, the replicated journal is used to rebuild the lost contents from the failed Storage Controller.

All journal contents are periodically de-staged to SSD non-volatile storage. In the event of a power loss, the system's NVRAM components are used to safely store the journals, and allow the cluster to complete an orderly shutdown. When metadata is de-staged to SSDs, the metadata is protected with XDP, as well as other techniques that are designed to guard against SSD failures.

Secure distributed journaling provides the system with its capability to recover, even in the unlikely catastrophic event that communication between all Storage Controllers is lost. Each Storage Controller becomes a self-sustained metadata protector, and can be brought up and reconnected to the system, once communications are restored.

Due to the importance of journaling, the journal mechanism code is completely separate from any other software module. It is a standalone, simple software module that is designed to be highly resilient.

Independent Software and Hardware Modules

XtremIO's flexible architecture ensures that any software component can run on any Storage Controller in the system. Having this flexibility provides the utmost availability and resiliency, while maintaining optimal performance.

Any changes occurring in the hardware configuration dynamically changes the number of active software modules. Such dynamic changes guarantee that all available resources are being optimally used by the system. For instance, a cluster comprised of four Storage Controllers has double the throughput and IOPS of a cluster with two Storage Controllers. Another example is the System-wide Management module (SYM) that runs on one Storage Controller. In the event of a hardware component failure, the SYM activates and runs on a different Storage Controller without any user intervention. Once the failed hardware component is replaced, the cluster quickly returns automatically to its optimal availability and level of performance.

Another factor that enables XtremIO to shift resources around is the fact that all software modules are loosely coupled. There is no affinity between the software and a specific hardware server, nor is there an affinity between specific software instances. A Data module can receive requests from any Control module, and therefore responds to it as a transaction. There is no need to remember a transaction, and the next transaction is a separate transaction. This architecture is similar to Service Oriented Architecture (SOA).

Connectivity Redundancy

The connectivity aspect of XtremIO maintains communications redundancy to every system component (see Table 1 and Table 2).

Not only does every component have at least two paths provided for communications, but also the management communication is on a separate network from that of the data flow. Host I/O is carried out via Fibre Channel or iSCSI ports, whereas management of the cluster is performed via dedicated Ethernet management ports on each Storage Controller.

Such a design allows for separation of control from the I/O path. Monitoring on a different network provides the ability to correlate events and system health level, independent of load or I/O behavior.

Table 1. XtremIO Connectivity Redundancy

Redundancy	Comment/Best Practice
Each Storage Controller configured with FC ports has two Fibre Channel ports.	Connect each port to a separate SAN switch.
Each Storage Controller has a minimum of two iSCSI ports.	Connect each port to a separate SAN switch.
Each Storage Controller has two InfiniBand ports.	Each port is connected to an independent InfiniBand fabric, providing fault tolerance against InfiniBand component failures.
There are two InfiniBand Switches (when more than one X-Brick is in the system).	Each switch is connected to every Storage Controller and guards against InfiniBand Switch failure.
There are two InfiniBand interconnect cables (when only one X-Brick is in the system).	Redundant InfiniBand paths run between the two Storage Controllers.
Each disk array enclosure (DAE) has two redundant data paths.	The data path is composed of DAE Controller, DAE Row Controller and SSD port. Failure in the data path does not result in loss of service.
Each disk array enclosure (DAE) SAS controller module utilizes two SAS cables.	Redundant SAS paths ensure that SAS port failures or SAS cable failures do not cause service loss.
Each NVRAM has 2 flash banks.	Failure of a flash bank does not result in loss of the NVRAM component.

Table 2. XtremIO Failure Service Impact

Failure	Action	Service Impact
Fibre Channel or iSCSI port	Host multi-pathing software uses the remaining ports.	No effect
InfiniBand port	The system uses the remaining InfiniBand port for data transfers to/from Storage Controllers.	No effect
InfiniBand Switch	The system uses the remaining InfiniBand Switch for internal data transfers.	No effect
Storage Controller	The Storage Controller partner in the same X-Brick takes over responsibility for all data in the disk array enclosure.	No service loss. Some performance loss occurs, as less overall I/O processing capability remains active.
Ethernet	The XMS cannot communicate with a Storage Controller.	No effect on I/O or on performance. The system's data path communications remain online via InfiniBand. The array cannot be configured or monitored until connectivity is restored.
Power supply units of Storage Controller, DAE, and InfiniBand Switch	The system notifies the administrator of the failure(s). A replacement power supply unit can be installed without service impact.	No effect. Dual power supplies allow the components to stay online.
Loss of power in one circuit	The system notifies the administrator of the failure. The system remains operational on the secondary, redundant circuit.	No effect
Loss of power in both circuits	The system performs de-stage of journals to non-volatile storage, and performs an orderly shutdown.	No service until the power is restored.
SSD	The system notifies the administrator. Automated SSD rebuild occurs (for up to two simultaneous SSD failures per XDP group).	Some performance loss occurs until the rebuild completes, depending on the fullness and utilization of the array. Less full and less busy arrays exhibit less performance impact and faster rebuilds.

End-To-End Verification

Hardware Verification

An important aspect of any storage system is the receipt of verification, given at every step of the data path. The different hardware data protection verification mechanisms are described in Table 3.

During data transfers between components, a cyclic redundancy check (CRC) is generated by the sending hardware component, which is then verified by the receiver. For any data at rest (in memory and on SSD), an error-correcting code (ECC) and a CRC are generated upon a write-to-memory, and are then verified upon a read.

Table 3. XtremIO Hardware Data Protections Verification Mechanisms

Hardware Component	Verification Type
Data Transfers – Fibre Channel, Ethernet, InfiniBand, PCIe, SAS	Hardware-based CRC
Data at Rest in Memory	DRAM ECC
Data at Rest on SSD	SSD ECC, SSD CRC, XtremIO XDP

XtremIO uses standard x86 servers, interface cards, InfiniBand components and eMLC SSDs. Each of these components includes highly-mature and robust hardware verification steps.

XtremIO avoids custom hardware modules in the array, as custom hardware requires substantial engineering work to achieve the same levels of resiliency readily available in standard enterprise-proven components.

Cryptographic Data Fingerprint

In addition to each XtremIO component in the data path having its own data verification mechanism, XtremIO employs an independent data check that is beyond the design of other storage systems. Upon receipt of an I/O from a host, the XtremIO Routing module (R module) computes a unique cryptographic data fingerprint which is based on the contents received from the host.

The cryptographic fingerprint is unique, and can only be correlated to a specific data pattern chunk. This cryptographic fingerprint is leveraged by the array's content-based data placement algorithms, as well as by the Inline Deduplication process.

The entire library of fingerprints is maintained in the Storage Controller memory. The cryptographic data fingerprint is recalculated from the outbound data and compared with the original fingerprint every time data is read by a host. This guarantees that the original information that is received from the host is stored safely on SSDs, is not inadvertently changed, and is properly delivered back to the host upon request.

Separate Message Paths

The calculated fingerprint information travels to the Data module in a separate message and along a different path than that of the data itself. This separation ensures that no component can corrupt the data and fingerprint in the same manner while in transit, thus preventing undetected data corruption. In short, a fingerprint is calculated upon data entry to the system and then recalculated and compared on every read from the SSD and upon data transfer to the host. The specific fingerprint travels within the XtremIO system separately from the data itself, thus providing an efficient method of independent checking.

Fault Avoidance, Detection, and Containment

Service-Oriented Architecture

XtremIO avoids cascading failure scenarios such as those that may occur on shared memory systems. XtremIO is built from different services that communicate with each other. Each service has its own set of data structures. If a fault exists in a service or in data, it is contained to that service/data.

Storage systems with large, shared memory and data structures are inherently more vulnerable to software errors, and therefore need to expend more resources trying to prevent cascading failures. XtremIO leveraged service-oriented architecture from the start to build a scalable system that is more robust than the large monolithic architectures that are used for other high-performance storage systems.

Fault Detection

The System-wide Management module (SYM) continuously monitors and detects hardware and software faults in the system. It continuously monitors Storage Controllers, disk array enclosures (DAEs), Fibre Channel HBAs, Ethernet NICs, InfiniBand HCAs, and InfiniBand Switches. The SYM also continuously monitors the SCSI driver, HBA controller drivers, Linux kernel, and battery communication software components.

Every component and every data path used in the XtremIO system has its own error detection method (as described in End-To-End Verification). For example, the eMLC SSDs in XtremIO have an LBA-seeded 32-bit CRC that is used for ECC miscorrect detection and on-the-fly correction. The SSDs are also equipped with 22-bit correction for every 512-byte sector and hardware-based RAID-5 within each SSD itself, to protect against internal flash module failures. This is separate from and in addition to XtremIO's XDP technology, and adds orders of magnitude in greater resiliency than typical in consumer MLC (cMLC) SSDs.

Advanced Healing

The SYM restarts failed software components automatically (as described in Platform Manager Module) and, upon hardware failure, can also reallocate software to different Storage Controllers. For example, if the SYM recognizes that the service to accept I/O from hosts (the R module) is not running, it restarts the service automatically. This capability ensures the utmost availability and optimized service levels, at all times.

The XtremIO array identifies unexpected data differences resulting from the fingerprint check that is performed upon reading from the SSD. XtremIO automatically rebuilds the missing data from all possible sources upon detection of such an inconsistency. This can be as simple as re-reading the data from the SSD in case the issue is transient. If the system is unable to read the data (or if the re-read process also produces incorrect results), the array rebuilds the data from the other SSDs in the XDP redundancy group.

The journaling and metadata in the system are critical for recovery from catastrophic events. Due to the importance of such datasets, the journals are protected by CRC for every written block.

Non-Disruptive Upgrades

XtremIO is designed and built for continuous availability. New firmware code is occasionally provided to add functionality, improve existing functionality and/or performance, and/or to fix known issues.

There are two types of upgrades:

- XtremIO OS (XIOS) Upgrades
- Component Firmware and Linux Kernel Upgrades

Both upgrade types are carried out with the system online to the host, and without downtime.

XtremIO OS (XIOS) Upgrades

XtremIO system updates are usually limited to XIOS, and only serve to modify the executable code that runs in user space. XIOS code is upgraded by loading the new code into resident memory on individual Storage Controllers. It also instantaneously converts all Storage Controllers to run the new code. The system is entirely available during the upgrade.

Component Firmware and Linux Kernel Upgrades

Individual hardware components can be upgraded, one at a time. For example, a Fibre Channel HBA can be upgraded by setting it offline to the host, upgrading the firmware, and then bringing it back online to the host. Once that Fibre Channel HBA is online, the system can upgrade the next Fibre Channel HBA. By leveraging host multi-pathing in accordance with XtremIO best practices, no downtime or any unavailability occurs to the host. This is also true for any other firmware upgrade, whether it is the SAS controller, InfiniBand, SSD firmware, or other firmware component.

In some cases, the Storage Controllers' Linux kernel may need upgrading. This upgrade is performed in the same fashion as the firmware. The Storage Controllers are individually upgraded, one at a time, with no impact on availability.

Scale Up

When additional capacity is required, the XtremIO Storage System can be scaled-up by adding additional SSDs to existing X-Bricks.

When the system expands during scale-up, there is no need for movement of the existing data. New data will be able to utilize the new capacity.

Scale Up is carried out without any need for configuration or manual movement of volumes.

Online Cluster Expansion

When additional performance or capacity is required, the XtremIO Storage System can be scaled-out by adding additional X-Bricks. Multiple X-Bricks are joined together over a redundant, high-availability, ultra-low latency InfiniBand network.

When the system expands, resources remain balanced, and data in the array is distributed across all X-Bricks to maintain consistent performance and equivalent flash wear levels.

System expansion is carried out without any need for configuration or manual movement of volumes. XtremIO uses a consistent fingerprinting algorithm that minimizes re-mappings. A new X-Brick is added to the internal load balancing scheme and only the relevant existing data is transferred to the new DAE.

System Recoverability

XtremIO shutdowns and power up processes are risk-free, due to the simple design of the system.

A shutdown is a graceful process that is initiated due to an external power loss, or upon user request. During shutdown, the journals are safely stored in the NVRAM component.

Each Storage Controller can shut itself down while remaining persistently capable of maintaining consistent data in case of a catastrophic event, such as cluster's communications becoming inadequate (for any reason), until power is restored.

Each Storage Controller has a local NVRAM component. Upon loss of communications or power, the journals are dumped to the NVRAM. Once power returns and communications are restored, the Storage Controller reconciles its journal information, along with the rest of the system.

XMS Communications Loss

The XtremIO Management System (XMS) is the application used to manage the XtremIO system. The XMS provides the graphical, command line and programmatic interfaces that are required to configure, provision, and monitor the system.

The system is managed via the dedicated Ethernet management ports. However, the system remains functional even if communication to the XMS is lost. The internal SYM is part of the array and runs on the Storage Controllers. I/O is continuously served, hardware remains monitored and any failed SSD initiates a rebuild and returns to complete redundancy. The only activities that are stopped are any user-initiated activities and monitoring (such as the creation of volumes), but host I/O and customer applications are not impacted.

Communication Loss between Storage Controllers

XtremIO Storage Controllers work in a loosely coupled architecture. Each Storage Controller works as a service to the other Storage Controllers, orchestrated by the SYM. However, each Storage Controller is an independent entity with the ability to protect the data it stores and maintain consistency when communications to other Storage Controllers are lost.

Upon the loss of communication to all Storage Controllers, the Storage Controller writes all metadata and journal information to the NVRAM and halts its services. This is similar to an Emergency Shutdown procedure. When communication is resumed and the Storage Controller becomes part of the system again, the Storage Controller reconciles the journal data, and once again becomes a system resource. The SYM module then reintegrates it back into the system and uses it as an I/O, caching and data facility. In the case of a single X-Brick, the remaining Storage Controller continues to serve I/O.

Conclusion

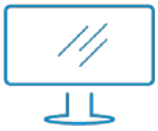
The XtremIO system's hardware and software design represents a leap forward in storage array technology, enabling new capabilities for most mission-critical workload consolidations, such as databases, analytics, and business applications. Multiple tiers orchestrated together achieve system high availability, data integrity, and data protection across all system components. The hardware provides redundancy for every host connection and path to data at rest. The XIOS operating environment provides a robust protection throughout the array's software stack, through fingerprint generation on data entry, separate paths for data and metadata, and a modular software design built in a service-oriented architecture. The different software modules run independently, yet act as a unified system. The overall system management is coherent, redundant, and can instantiate software modules on different hardware components.

XtremIO achieves greater than 99.999% high availability, data integrity and data protection by employing the following features:

- Hardware redundancy for every component
- Unique content fingerprinting as data is written
- Separate paths from system entry to SSDs for user data and its accompanying fingerprint
- Secured journaling to protect against unexpected system shutdowns, component failures or communication failures
- Loosely-coupled software modules working together in a service-oriented architecture
- Centralized redundant management
- Redundancy against up to two simultaneous SSD failures per XDP group
- Non-disruptive system software upgrades

How to Learn More

For a detailed presentation explaining XtremIO Storage Array's capabilities and how XtremIO substantially improves performance, operational efficiency, ease-of-use, and total cost of ownership, please contact XtremIO at XtremIO@emc.com. We will schedule a private briefing in person or via a web meeting. XtremIO provides benefits in many environments and mixed workload consolidations, including virtual server, cloud, virtual desktop, database, analytics, and business applications.



Learn more about Dell EMC XtremIO

Contact a Dell EMC Expert

View more resources

Join the conversation @DellEMCStorage and #XtremIO