

Table of contents

- [1:Overview](#)
- [2:CORTX-Cluster-R2](#)
 - [2.1:Storage-Set](#)
 - [2.1.1:Enclosure](#)
 - [2.1.1.1:Enclosure-SSD-Config-For-Metadata](#)
 - [2.1.2:Server](#)
 - [2.2:Failure-domain](#)

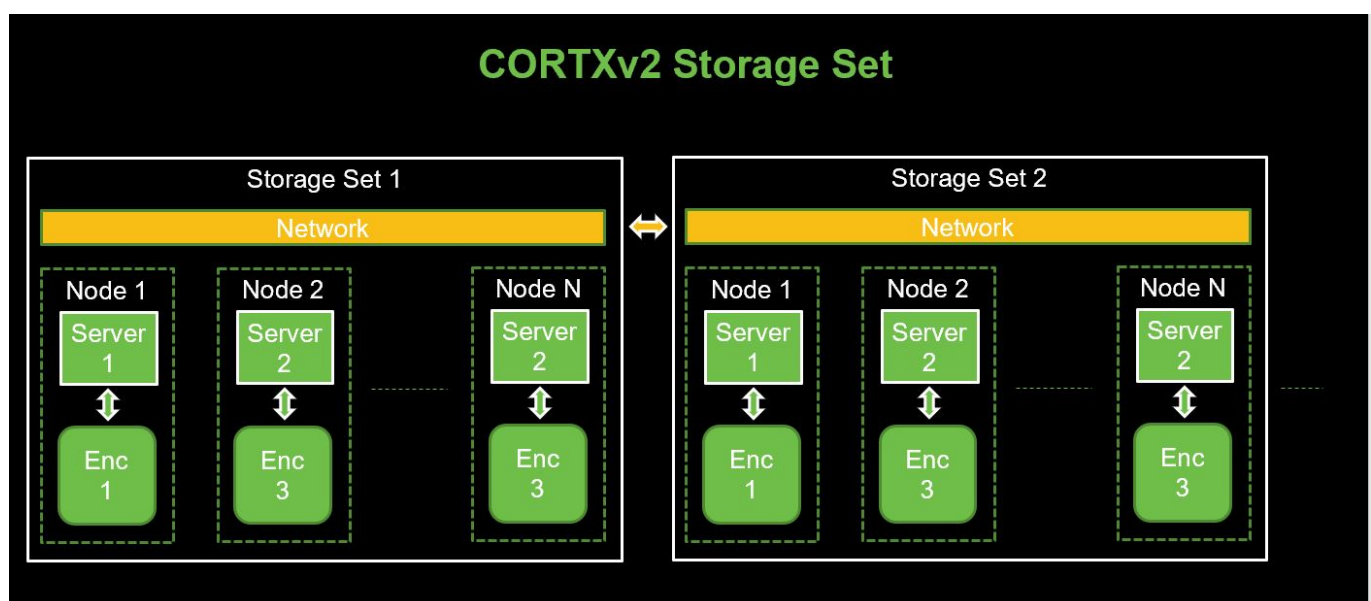
1:Overview

This document will be describe LDR R2 motr architecture and design

2:CORTX-Cluster-R2

Following are the hierarchy of Cortx Cluster

1. Cortx Cluster
 - Top Level cluster will have multiple storage set
 - Each storage set will be connected at network layer
2. Storage Set
 - Storage set will have multiple nodes
 - Number of nodes can vary with storage set (e.g $N1 \neq N2$)
 - The capacity of storge set can differ even if number of nodes are same ($N1 == N2$)
3. Nodes
 - Node consist of server and enclosure pair.
 - For R2, enclosure is isolated and only node can access its enclosure.



Assumption: CORTX R2 will have following characteristics

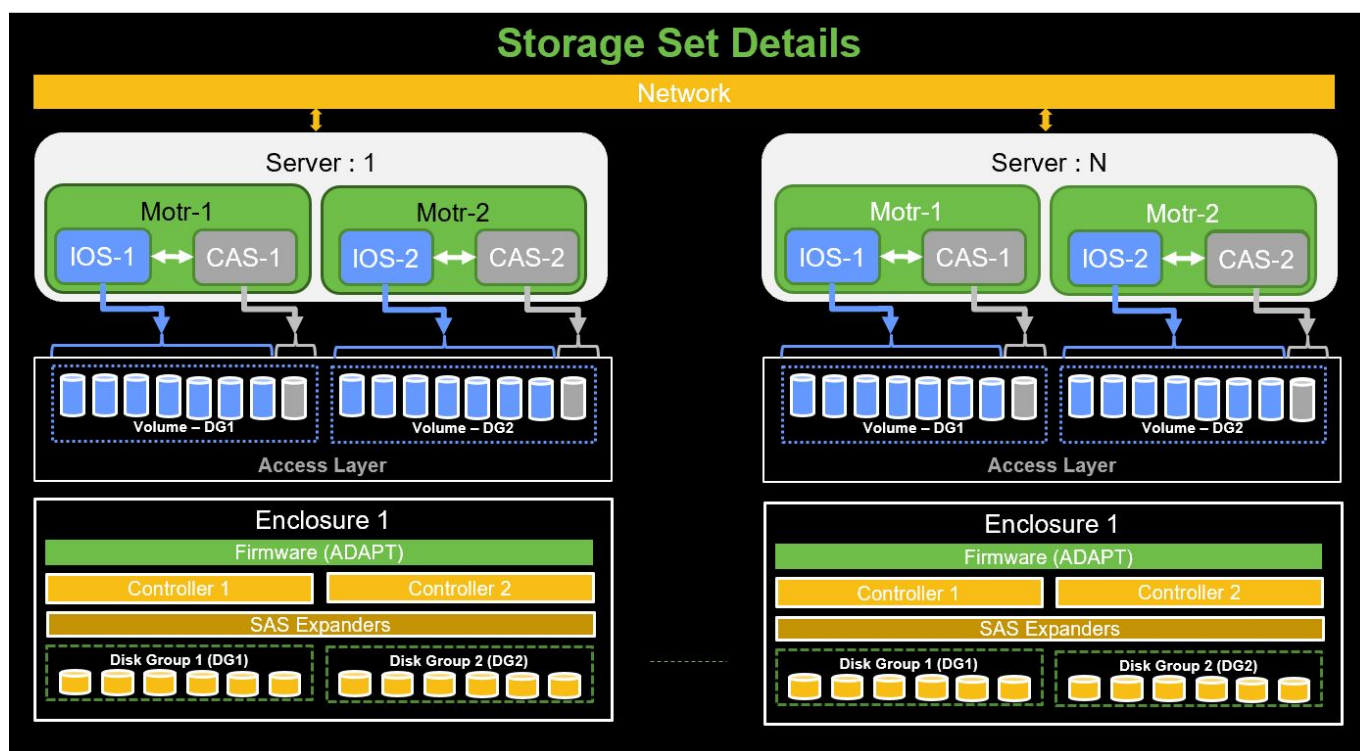
1. Storage set will communicate with each other at network layer.
 - e.g In above figure Storage Set 1 and 2 are shown connected at network layer
2. **IOs will be confined to Storage Set.**
 - Data will only be striped across nodes in a Storage Set.
3. Any node in cluster can server S3 request for any storage set.
 - e.g In above figure Node 1 from Storage Set 1 can server read request for object stored in Storage Set 2

Note that any change in the above assumption will have implications on design.

2.1:Storage-Set

The figure below shows the details of storage set

- Storage set consist of N number of nodes i.e server and enclosure pair
- For LDRR2 each enclosure will have two Disk Droup (DG) and DG can be a failure domain.
- Two serve two DG, each server will need two instance of motr IO and CAS running on it.



Elements of Storage set are described in following section

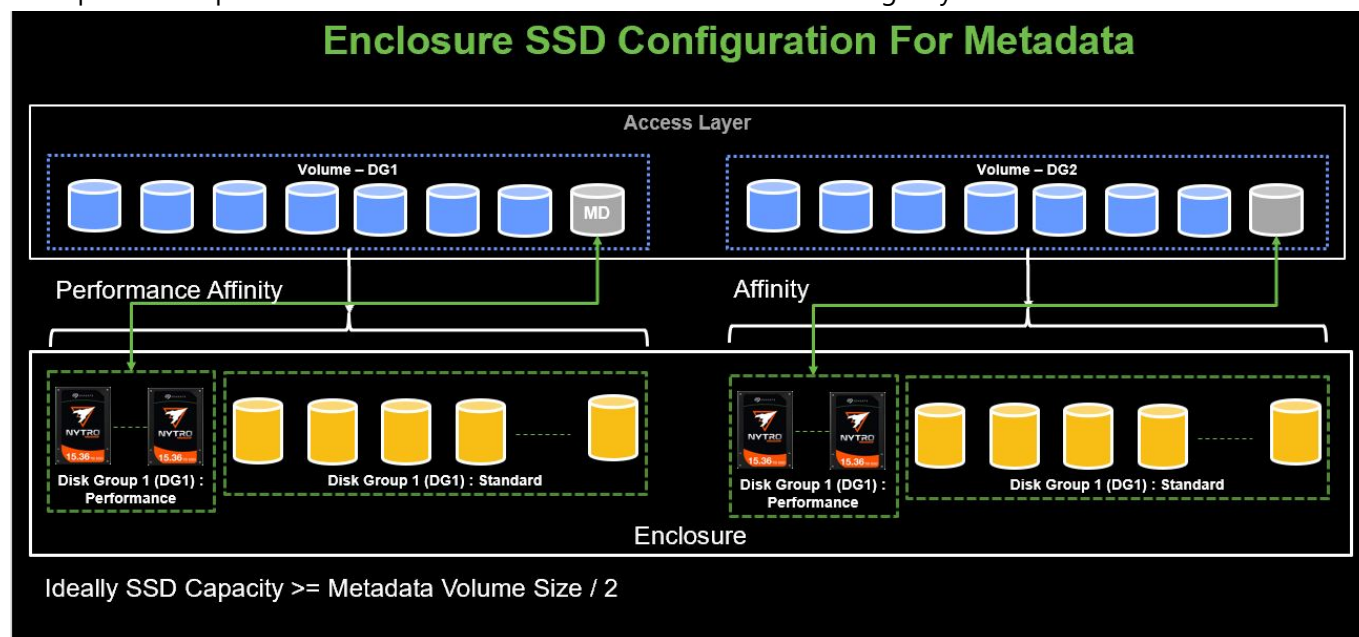
2.1.1:Enclosure

Enclosure+Firmware creates Disk Group (DG) by grouping set of drives which controller can access. Controller access drives through SAS Expander, so each controller can access all DG. This helps to provide controller failover.

Enclosure Firmware also provides management interface to create set of volumes. As per enclosure team creating 8 volume gives optimal performance

2.1.1.1:Enclosure-SSD-Config-For-Metadata

To improve the speed of metadata access SSDs can be used in following ways in enclosure.



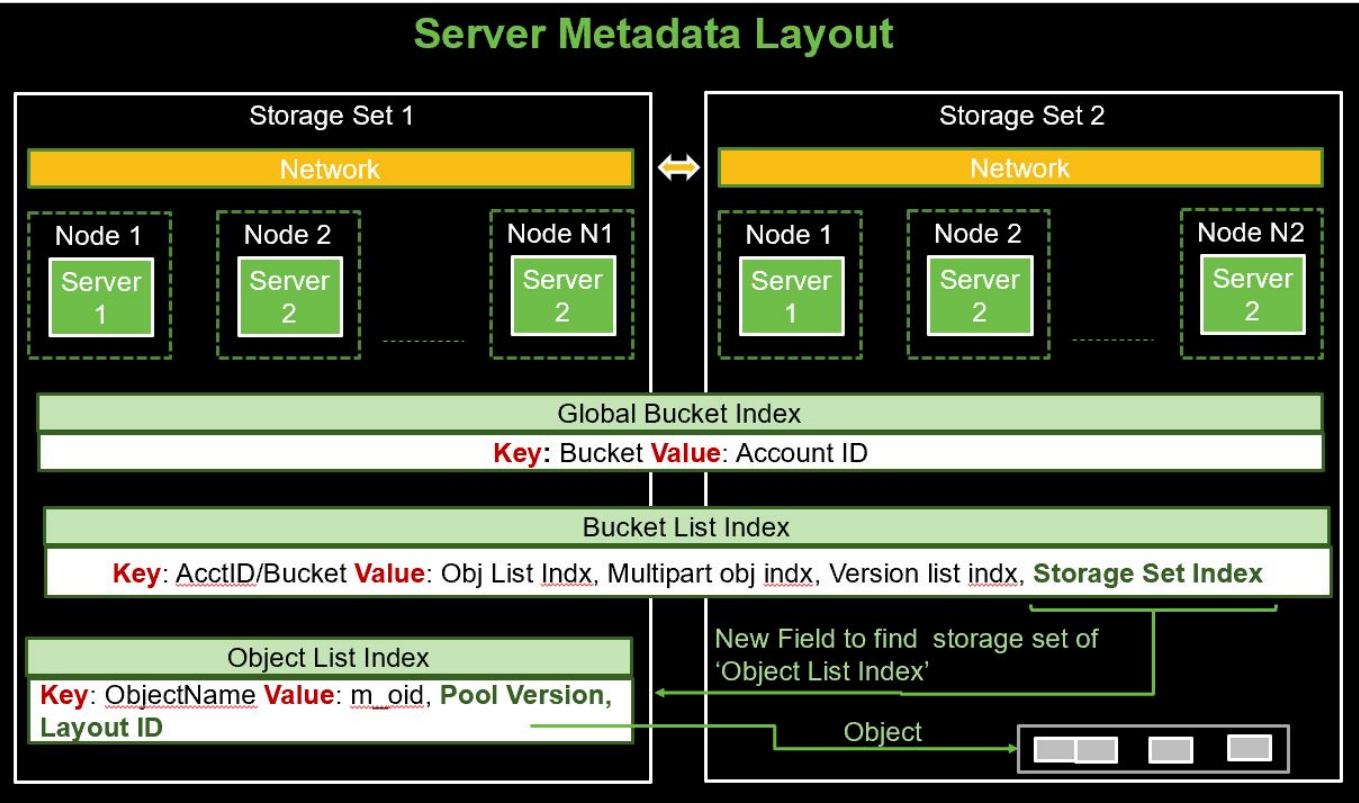
- HDDs will be part of Standard Disk Group while
- SSDs will be part of Performance Disk Group
- Metadata volume affinity should be set for Performance
- Ideally SSD Capacity \geq Metadata Volume Size / 2

TODO: Analyze the impact of multiple DG in a storage pool and it's failure handling

2.1.2:Server

All the Servers in CORTX cluster are able to communicate with each other using network interface. As S3 request can arrive at any node in the cluster, S3 server (running on node) should be able to locate the storage set in which object info is located and then the further details about object can be retrieved. **Note:** Object info

and Object can reside on different storage set.



To support the above functionality following S3 metadata should be replicated across storage set

- 1. Global Bucket Index
- 2. Bucket List Index

As the size of this metadata is small, it will be beneficial to replicate this across all the nodes of cluster and during lookup, the local node can provide all the data. Caching mechanism will also help to speedup the lookup for these data.

Note that following new fields needs to be added to existing S3 metadata:

- New field needs to be added to the Bucket List Index metadata to locate storage set in which 'Object List Table' is present.
- 'Object List Table' to storage 'Pool Version' and 'Layout ID' locate the Storage Set in which object is located.

JIRA: S3 metadata should have version field and reserved field to allow upgraded to metadata without any need for migration?

** NOTE: Work In Progress **

2.2:Failure-domain

This section will analyze failure domain and its impact on data/metadata consistency. With respect of motr following category of failure can occur:

Failure	Failure Component	Description	Motr Failure Domain Mapping
Storage	Disk Group	Single/Dual Disk Group Failure	DG Failure

Failure	Failure Component	Description	Motr Failure Domain Mapping
	Controller	Single Controller Failure	No Impact (User notification)
		Both Controller Failure	Dual DG Failure / Node Failure
	SAS HBA	Both DG inaccessible	Dual DG Failure / Node Failure
	Enclosure	Both DG inaccessible	Dual DG Failure / Node Failure

Network

Failure	Failure Component	Description	Desired Action
Network	Network Interface	Single Node getting isolated from others	Majority node which are part of storage set will make quorum and single node to be taken down by HA action
	Controller	Single Controller Failure	No Impact (User notification)
		Both Controller Failure	Dual DG Failure / Node Failure
	SAS HBA	Both DG inaccessible	Dual DG Failure / Node Failure
	Enclosure	Both DG inaccessible	Dual DG Failure / Node Failure

1. Network Unit :

- NW Interface failure
- NW Partion

Failure	Failure Type	Side Effect	Desired Action
Storage	Temporary	IOs will timeout for that DG/Volume	-Total IO timeout should be less than S3 timeout
			-Method to detect storage unit unavailable to avoid sending IO request to it
			-Method to detect storage unit is available to allow sending IO request to it

1. Storage Unavailable a. Temporary Failure b. Permanent Failure

2. Network Unavailable a. Temporary Failure b. Permananet Failure

Basic Assumptions

Following are the basic assumptions for creating the sequence flow for various error scenarios

- Global Bucket list is replicated across storage set. -- Lookup into global metadata will give Storage Set where the bucket list and data reside
- Bucket Object List Table is replicated within storage set -- Bucket Object List Table if we make this global than size of global data increases lot? So this should be within storage set, but objects can be part of different storage set but Object data does not span across storage set ?

Initial Condition :

- Two Storage Set (SS) in cluster
- Each storage set with 3 nodes
- Single failure in a node is allowed
- Global bucket is replicated across all nodes of cluster (N):
 - This is small amount of global bucket data
 - Frequency of creation of this bucket data is low
 - Replication to all nodes allows each storage set to easily support one node failure and will also help with case of Storage Set addition. -- Full replication will help in performance, minimum replication needed in a storage set is $\# \text{num_of_node_failure_in_storage_set} + 1$
- Local S3 metadata should be replicated across atleast 3 nodes of a storage set
 - Replication of data across 3 nodes of storage set, will helps in node failure
- Data is striped in 4+2 parity config

DTM Usage

This section will analyze the DTM usage in various scenario with assumption that

Scenario

Following scenario will be analyzed w.r.t. DTM role to restore storage system to consistent state w.r.t metadata and data.

- IO Failure: Node or IO Service going down temporarily
- Software Upgrade: Node going through SW upgrade.
- Storage Set Addition: A new storage set becoming part of cluster

IO Failure

A node or storage unavailable for some time can cause following issues

1. Node will not be able to update following metadata and it will get out of sync with cluster
 1. Global Bucket List : Across Storage Set
 2. Bucket Object List Table : Within Storage Set
2. Data unit directed to the node will get dropped and object will lose units of data (1-MAX_DISK_GROUP_IN_SS)

Software (SW) Upgrade

Without DTM, cluster can service Read Object request during SW upgrade but for write object and bucket creation metadata will need DTM.

Storage Set Addition

A new storage set addition will need to sync Global Bucket List and DTM will be needed for that.

As for P0 we can live with object in degraded state, we can skip DTM for Data. But the DTM is mandatory for keeping the cluster distributed metadata in consistent state.

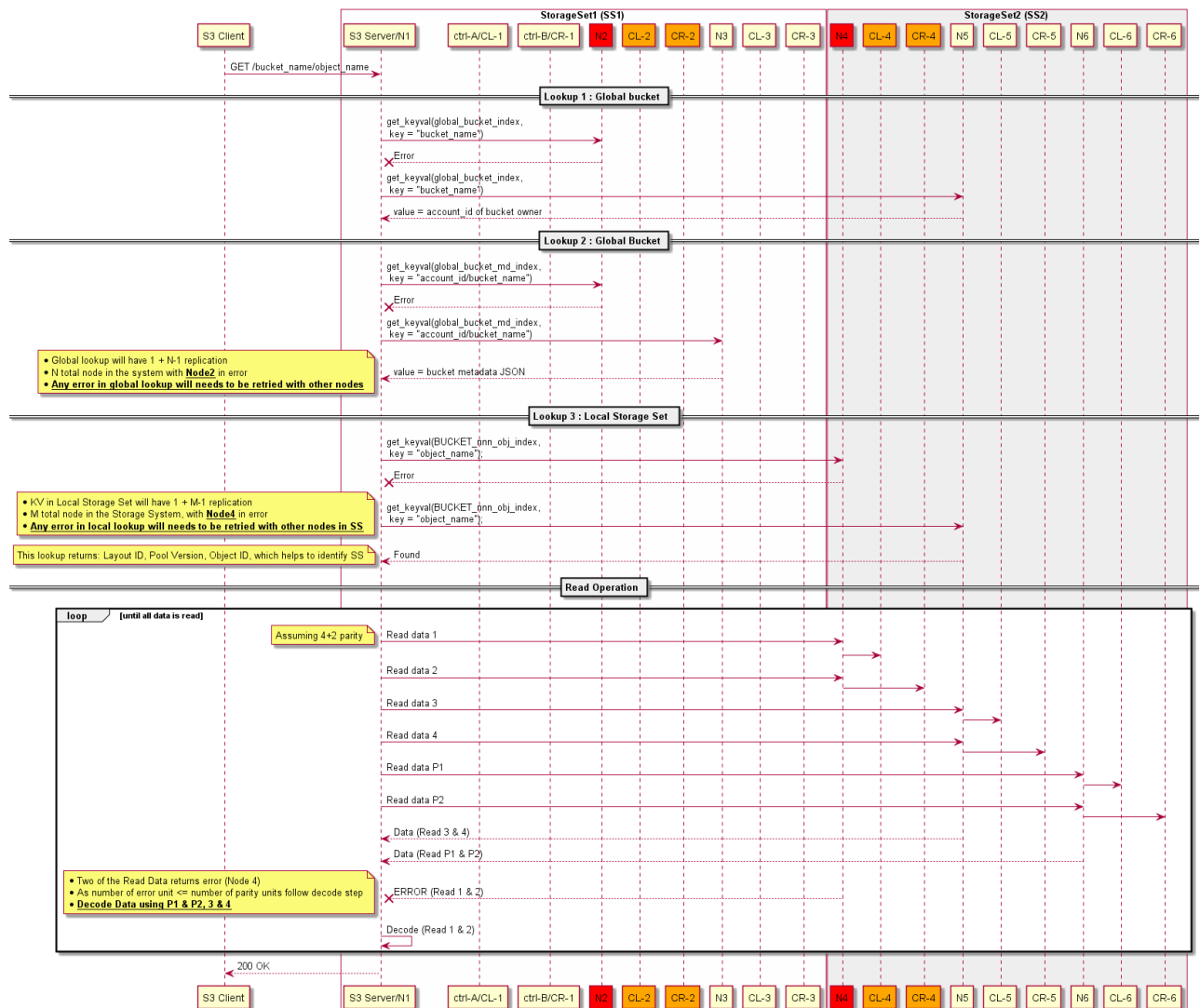
I. Node failure During IO

Error Scenario :

- One node from each storage set (SS) fails during IO.
- Any error in reading metadata should be retried with metadata available with replicated node

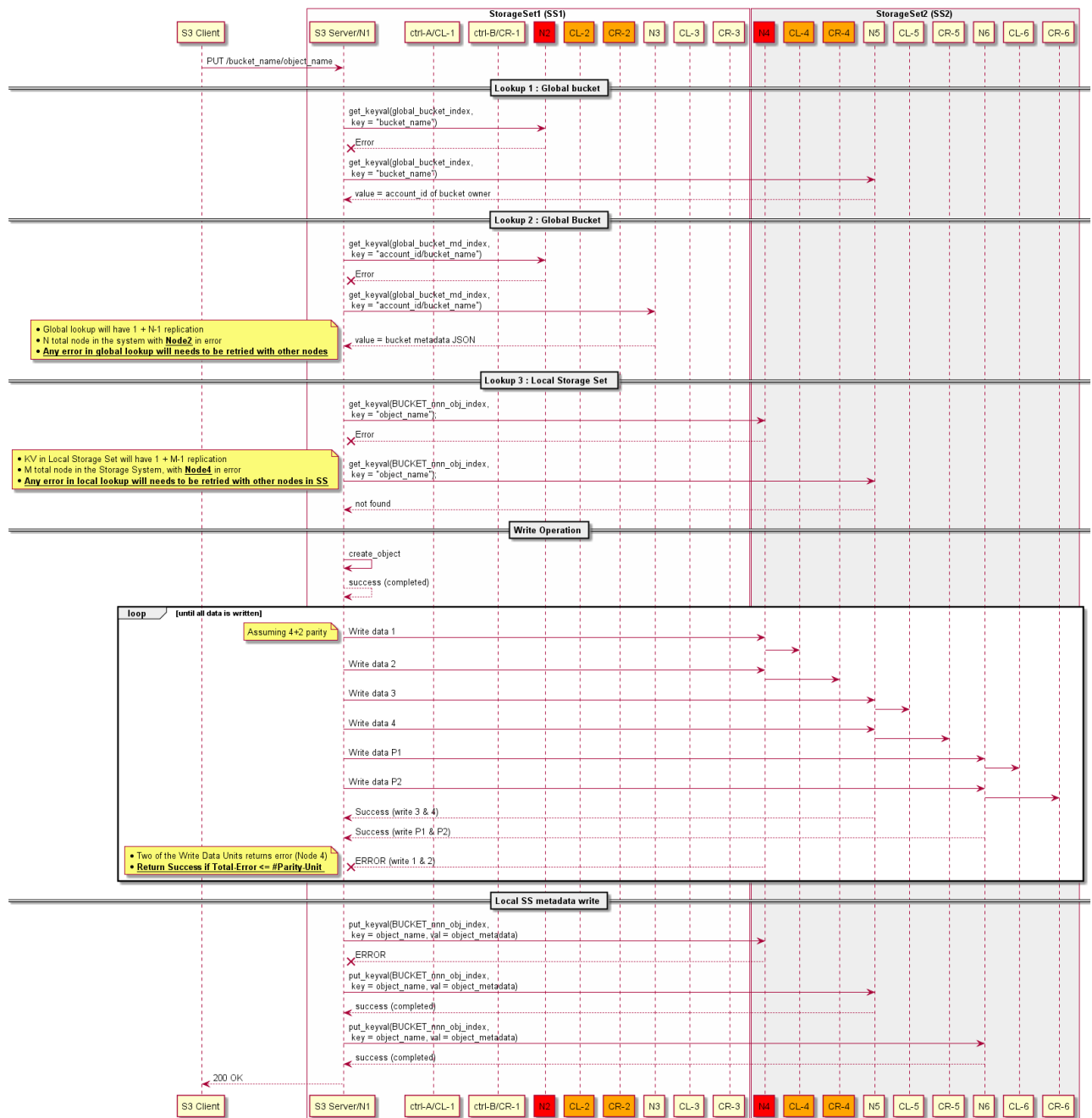
1. Simple Object Get/Read

- Read path should detect error in retrieving data and should use parity units to get missing data.



2. Simple Object Put/Write

- Write path should detect error in writing data and should return success if the failed number of write unit is less than or equal to number of parity unit



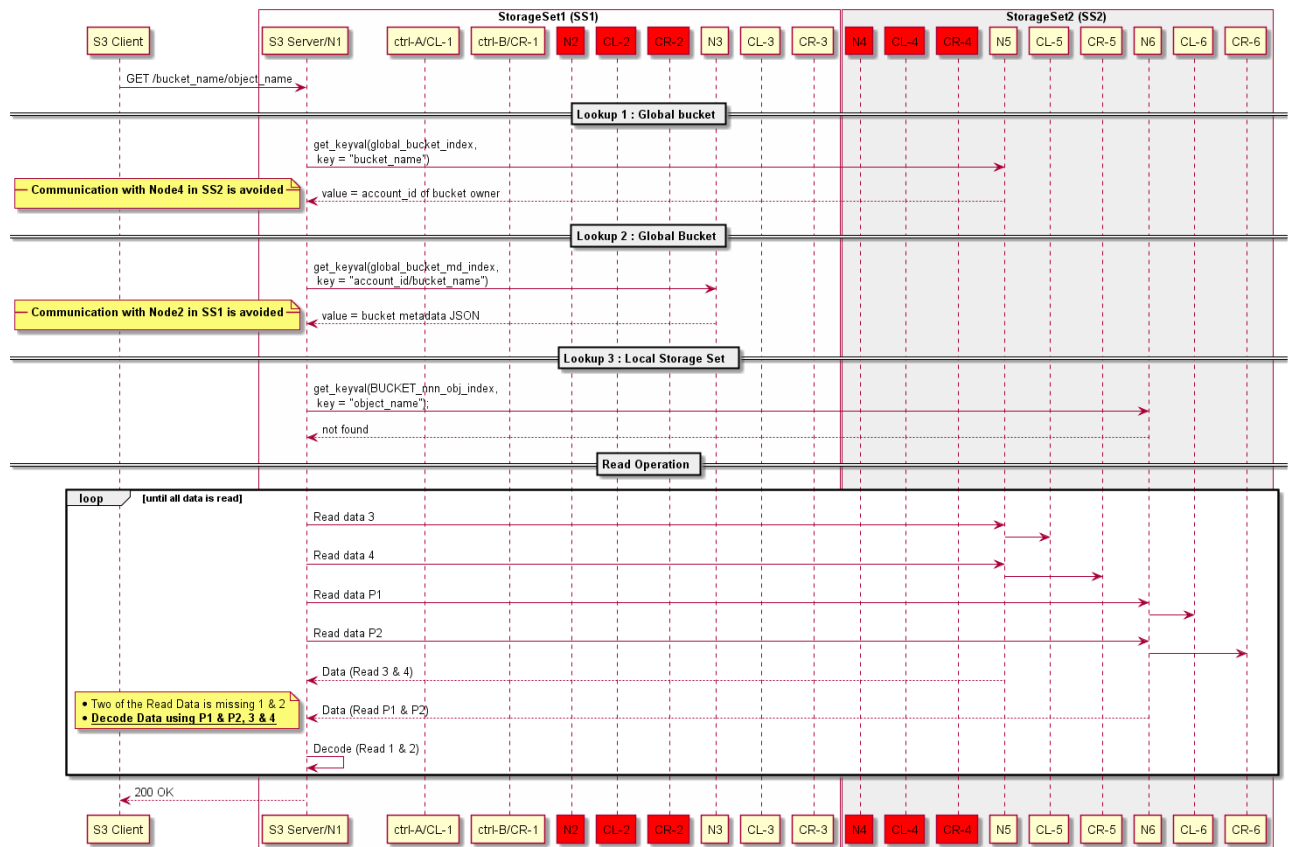
II. Detected Node failed Scenario

Error Scenario :

- One node from each storage set (SS) has failed.
- Motr is aware of failure and Hare has notified to motr about the node failure

1. Simple Object Get/Read

- Read path error handling avoids communicating with failed node for metadata and data operations.



2. Simple Object Put/Write

- Write path error handling avoids communicating with failed node for metadata and data operations.



