IBM Spectrum Scale Concepts and features

—

Tomer Perry
Spectrum Scale development
<tomp@il.ibm.com>

Outline

- What is a filesystem?
- Filesystem types
- What is IBM Spectrum Scale?
- Basic functionality and constructs
- Advanced functionality

What is a File System ?

In computing, a file system (or filesystem) is used to control how data is stored and retrieved. Without a file system, information placed in a storage area would be one large body of data with no way to tell where one piece of information stops and the next begins. By separating the data into pieces and giving each piece a name, the information is easily isolated and identified. Taking its name from the way paper-based information systems are named, each group of data is called a "file". The structure and logic rules used to manage the groups of information and their names is called a "file system" [(1)]

[1] http://en.wikipedia.org/wiki/File_system

What is a File System ?

In computing, a file system (or filesystem) is used to control how data is stored and retrieved. Without a file system, information placed in a storage area would be one large body of data with no way to tell where one piece of information stops and the next begins. By separating the data into pieces and giving each piece a name, the information is easily isolated and identified. Taking its name from the way paper-based information systems are named, each group of data is called a "file". The structure and logic rules used to manage the groups of information and their names is called a "file system" [1]

<u>Basic Features:</u>

- Space Management
- Namespace (files, dirs)
- Metadata
- Utilities (std. vs. non std.)

<u>Value Add:</u>

- Access control (*)
- Maintaining integrity (*)
- Scalability
- Availability

[1] http://en.wikipedia.org/wiki/File_system

What is a File System ?

In computing, a file system (or filesystem) is used to control how data is stored and retrieved. Without a file system, information placed in a storage area would be one large body of data with no way to tell where one piece of information stops and the next begins. By separating the data into pieces and giving each piece a name, the information is easily isolated and identified. Taking its name from the way paper-based information systems are named, each group of data is called a "file". The structure and logic rules used to manage the groups of information and their names is called a "file system" (1)

Basic Features:

- Space Management
- Namespace (files, dirs)
- Metadata
- Utilities (std. vs. non std.)

Value Add:

- Access control (*)
- Maintaining integrity (*)
- Scalability
- Availability and monitoring

Beyond a simple filesystem:

- Parallel/shared
- High performance
- Highly scalable
- ILM/HSM

- File and storage virtualization
- Snapshots
- Replication
- Encryption
- Compression
- Acceleration
- Geo Distribution
- Checksum
- Etc. etc. etc.

What is a clustered file system ?

A clustered file system is a file system which is shared by being simultaneously mounted on multiple servers. There are several approaches to clustering, most of which do not employ a clustered file system (only direct attached storage for each node). Clustered file systems can provide features like location-independent addressing and redundancy which improve reliability or reduce the complexity of the other parts of the cluster. Parallel file systems are a type of clustered file system that spread data across multiple storage nodes, usually for redundancy or performance(1)

[1] http://en.wikipedia.org/wiki/Clustered_file_system

What is a clustered file system ?

A clustered file system is a file system which is shared by being simultaneously mounted on multiple servers. There are several approaches to clustering, most of which do not employ a clustered file system (only direct attached storage for each node). Clustered file systems can provide features like location-independent addressing and redundancy which improve reliability or reduce the complexity of the other parts of the cluster. Parallel file systems are a type of clustered file system that spread data across multiple storage nodes, usually for redundancy or performance(1)

Shared filesystems

[1] http://en.wikipedia.org/wiki/Clustered_file_system

What is a clustered file system ?

A clustered file system is a file system which is shared by being simultaneously mounted on multiple servers. There are several approaches to clustering, most of which do not employ a clustered file system (only direct attached storage for each node). Clustered file systems can provide features like location-independent addressing and redundancy which improve reliability or reduce the complexity of the other parts of the cluster. Parallel file systems are a type of clustered file system that spread data across multiple storage nodes, usually for redundancy or performance[1]

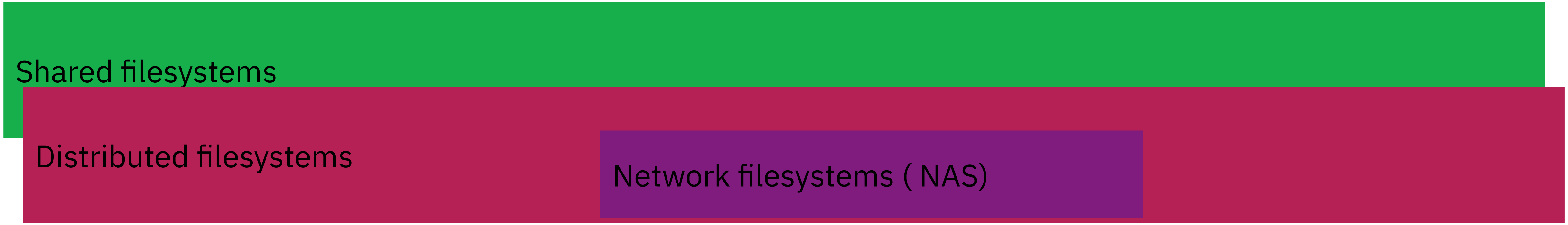
Shared filesystems

Distributed filesystems

[1] http://en.wikipedia.org/wiki/Clustered_file_system

What is a clustered file system ?

A clustered file system is a file system which is shared by being simultaneously mounted on multiple servers. There are several approaches to clustering, most of which do not employ a clustered file system (only direct attached storage for each node). Clustered file systems can provide features like location-independent addressing and redundancy which improve reliability or reduce the complexity of the other parts of the cluster. Parallel file systems are a type of clustered file system that spread data across multiple storage nodes, usually for redundancy or performance[1]



Shared filesystems

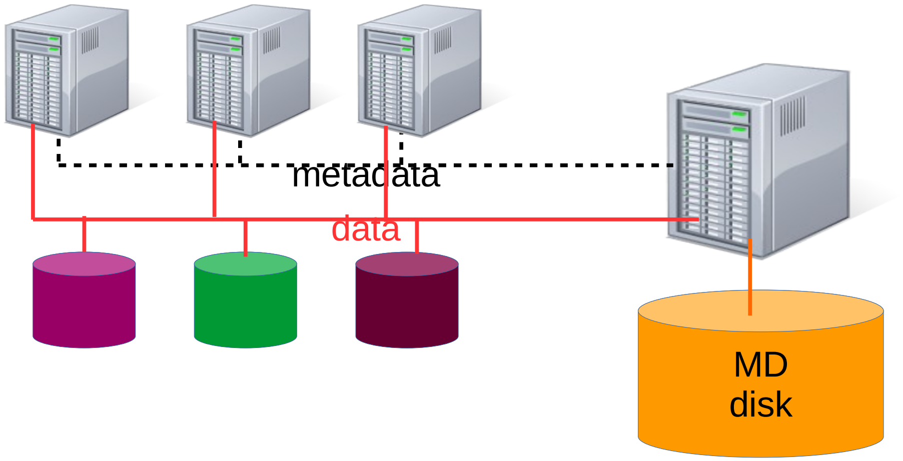
Distributed filesystems

Network filesystems (NAS)

[1] http://en.wikipedia.org/wiki/Clustered_file_system

What are shared file systems ?

- Shared file systems (a.k.a. Shared disks filesystems) usually separate the metadata and data paths
- The nodes usually have direct access to the shared disks (SAN)
- There is a “metadata server” (or clusters of) that centralize metadata activity



¹ http://en.wikipedia.org/wiki/Clustered_file_system

What are distributed file system ?

- Distributed file systems do not share block level access to the same storage but use a network protocol
- Most current implementations uses some network file protocol (NFS, CIFS etc.)

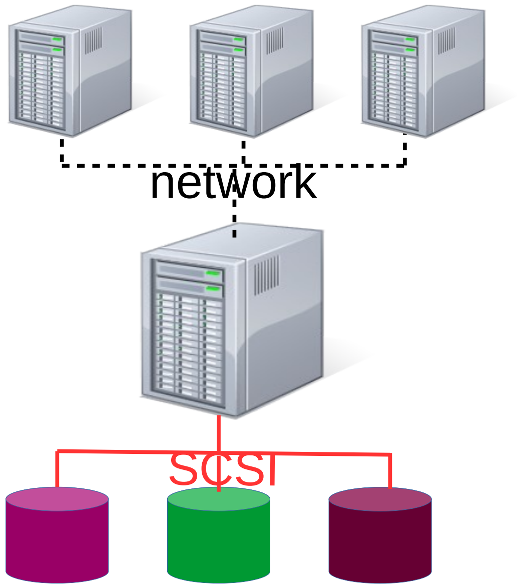
Single server

Pros

- Simple
- Cheap (?)

Cons

- Limited Scalability



What are distributed file system ?

- Distributed file systems do not share block level access to the same storage but use a network protocol
- Most current implementations uses some network file protocol (NFS, CIFS etc.)

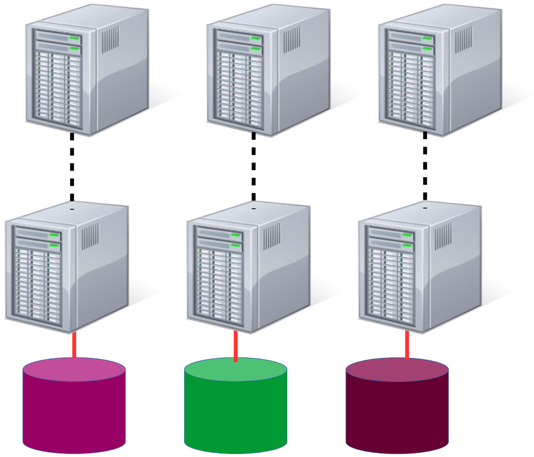
Clustered NAS (common)

Pros

- Relatively Simple
- Relatively Cheap
- Relatively Scalable

Cons

- Limited Scalability
- Not really parallel (1x1)



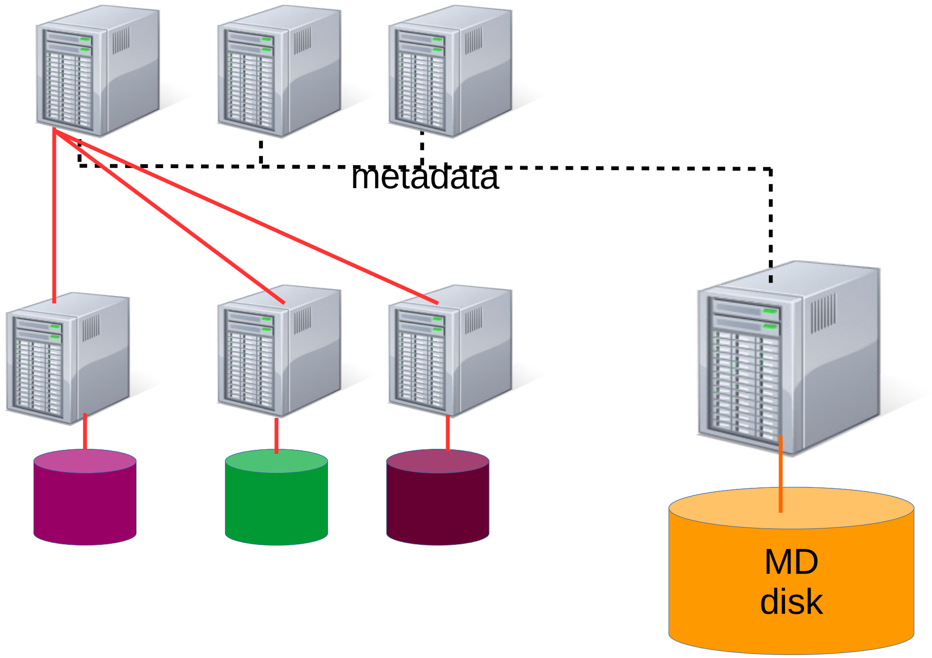
What are distributed file system ?

- Distributed file systems do not share block level access to the same storage but use a network protocol
- Most current implementations uses some network file protocol (NFS, CIFS etc.)

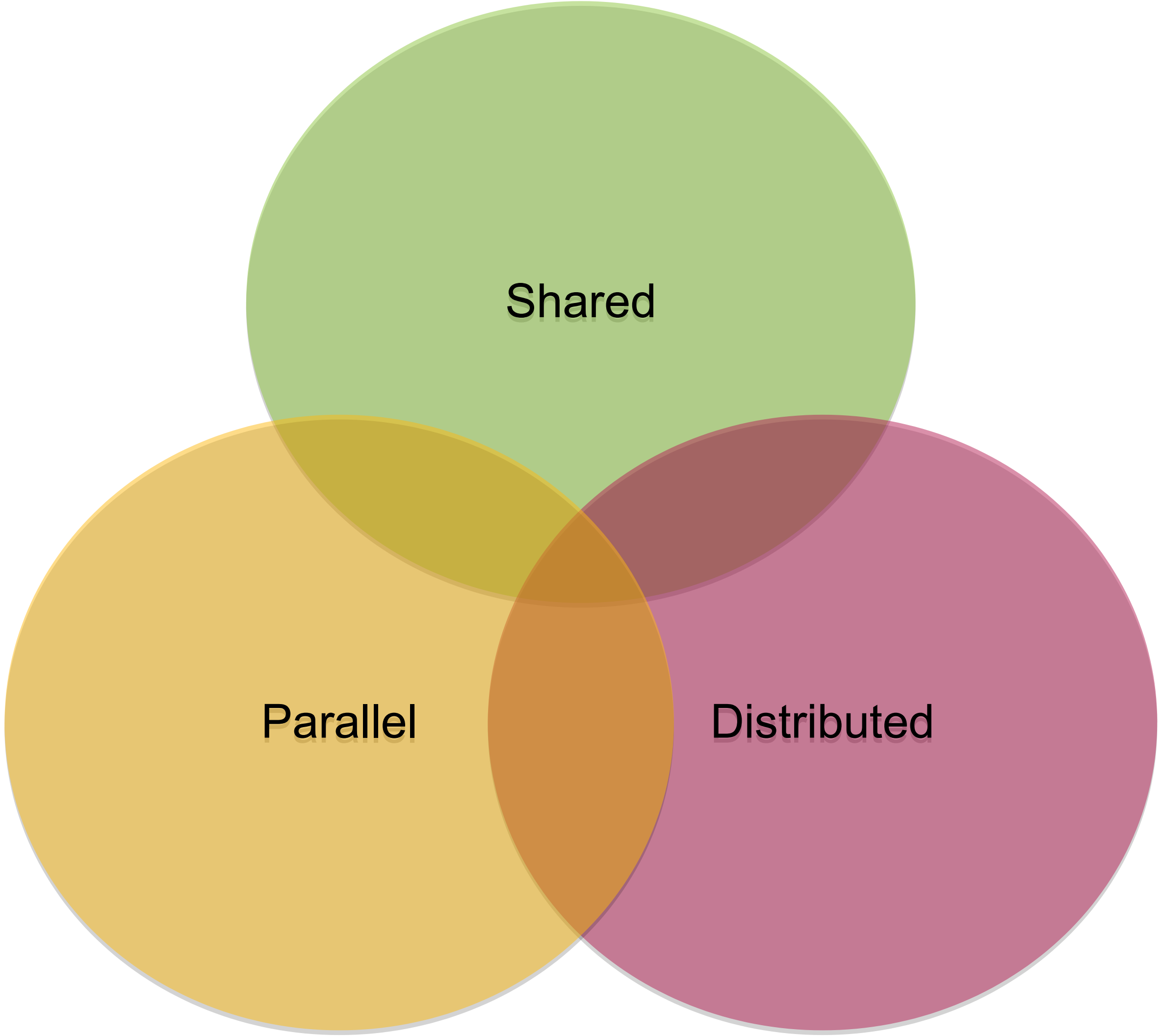
Parallel NAS (pNFS, SMB3)

Pros
- Relatively Simple
- Relatively Cheap
- Relatively Scalable

Cons
- Not commonly used (yet?)



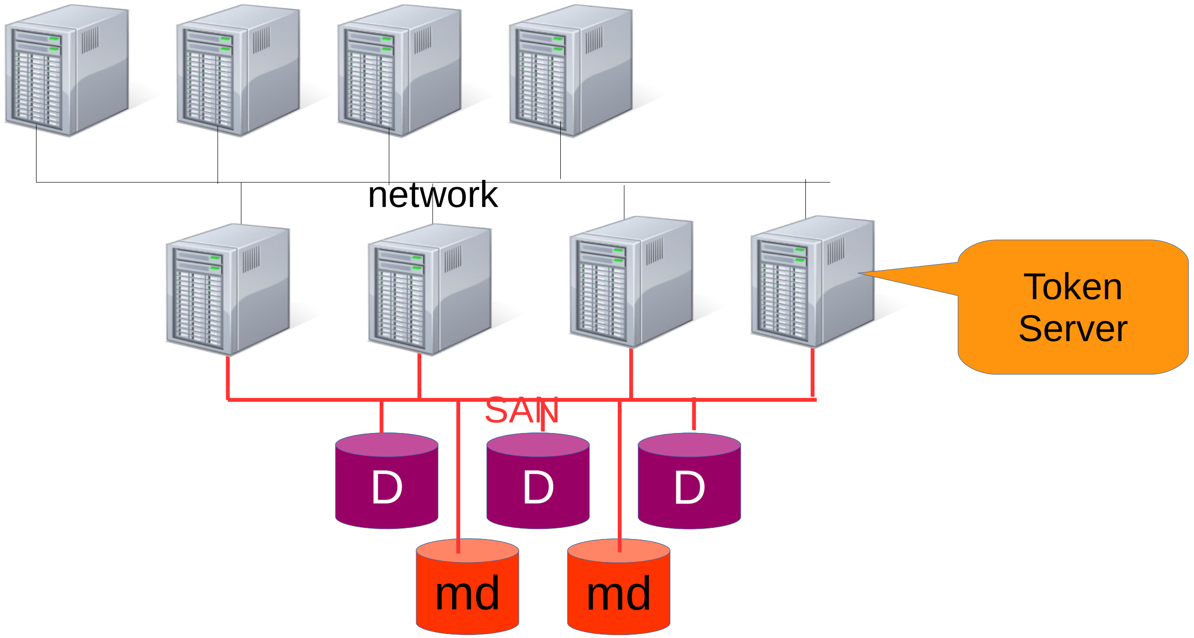
So, what is Spectrum Scale ?



So, what is Spectrum Scale ?

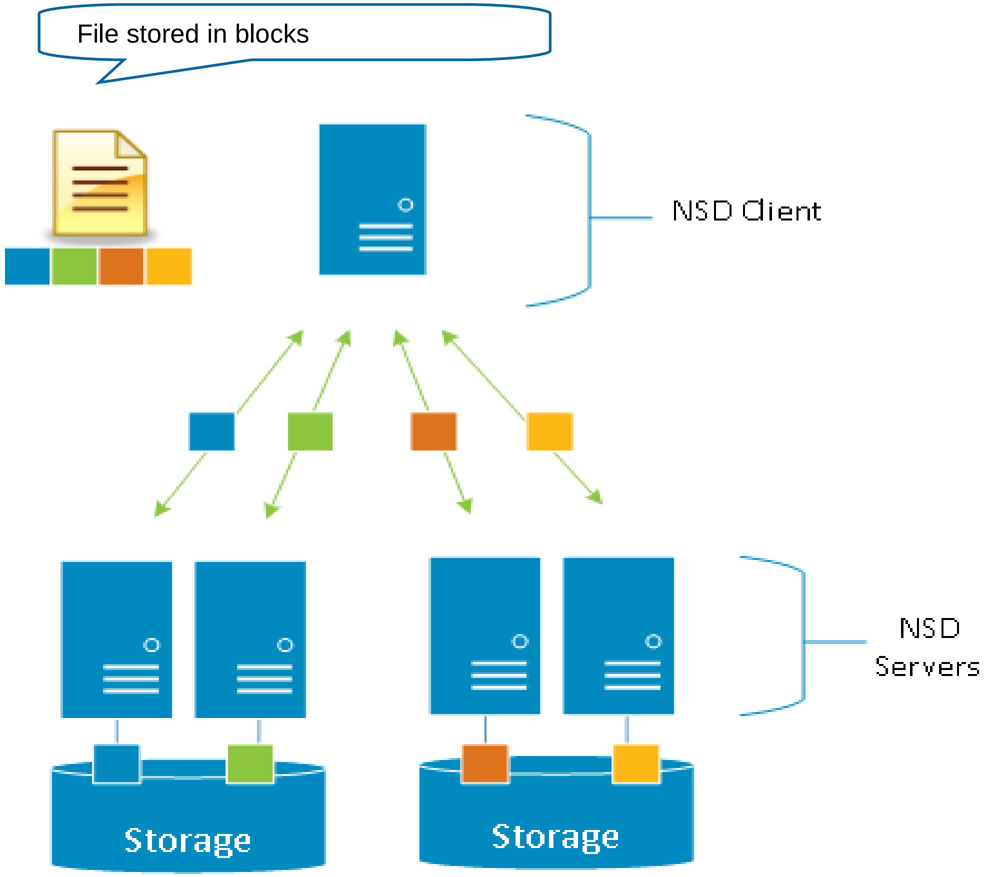
- Spectrum Scale took a different approach – using a token based lock manager in order to maintain consistency
- Every node has client to both data and metadata, using SAN shared disk, network access to shared disk (or combination of the two)

- Token server grants tokens
- Token represent the right to read, cache and/ or update piece of data or metadata
- Single message to token server allows repeated access to the same object
- Revoke on conflicting operation
- Force-on-steal: dirty data & metadata flushed to disk when token is stolen



Spectrum scale parallel architecture

- Clients use data, Network Storage Devices (NSDs) serve shared data
- All NSD servers export to all clients in active-active mode
- Spectrum Scale stripes files across NSD servers and NSDs in units of file-system block-size
- NSD client communicates with all the servers
- File-system load spread evenly across all the servers and storage. No HotSpots
- Easy to scale file-system capacity and performance while keeping the architecture balanced



NSD Client does real-time parallel I/O to all the NSD servers and storage volumes/NSDs

Token based locking - optimizations

- Byte range locking
  - Opportunistic locking: small “required” and large “requested” - minimize lock traffic
- Metadata token optimizations
  - Acquire ability to open on lookup
  - Inode reuse: already have token
  - Token prefetch on readdir
  - “metanode” dynamically assigned to handle metadata updates on write sharing
- Block allocation map
  - Segmented allocation map allows each node to allocate space on all disks with minimal coordination
- Special tokens for DIO, fastpath for “safe” access
- And…many others

Spectrum Scale Cluster

- In order to monitor nodes health, manage distributed resources etc. - Spectrum Scale is using a built in clustering infrastructure
- Apart from quorum, all the nodes are identical – where Scale chooses several nodes in order to perform centralized tasks

Config Servers

- Holds cluster configuration files
- 4.1 onward supports CCR
- Not in the data path

Cluster Manager

- One of the quorum nodes
- Manage leases, failures and recoveries
- Selects filesystem managers
- mm*mgr -c

Filesystem Manager

- One of the “manager” nodes
- One per filesystem
- Manage filesystem configurations (disks)
- Space allocation
- Quota management

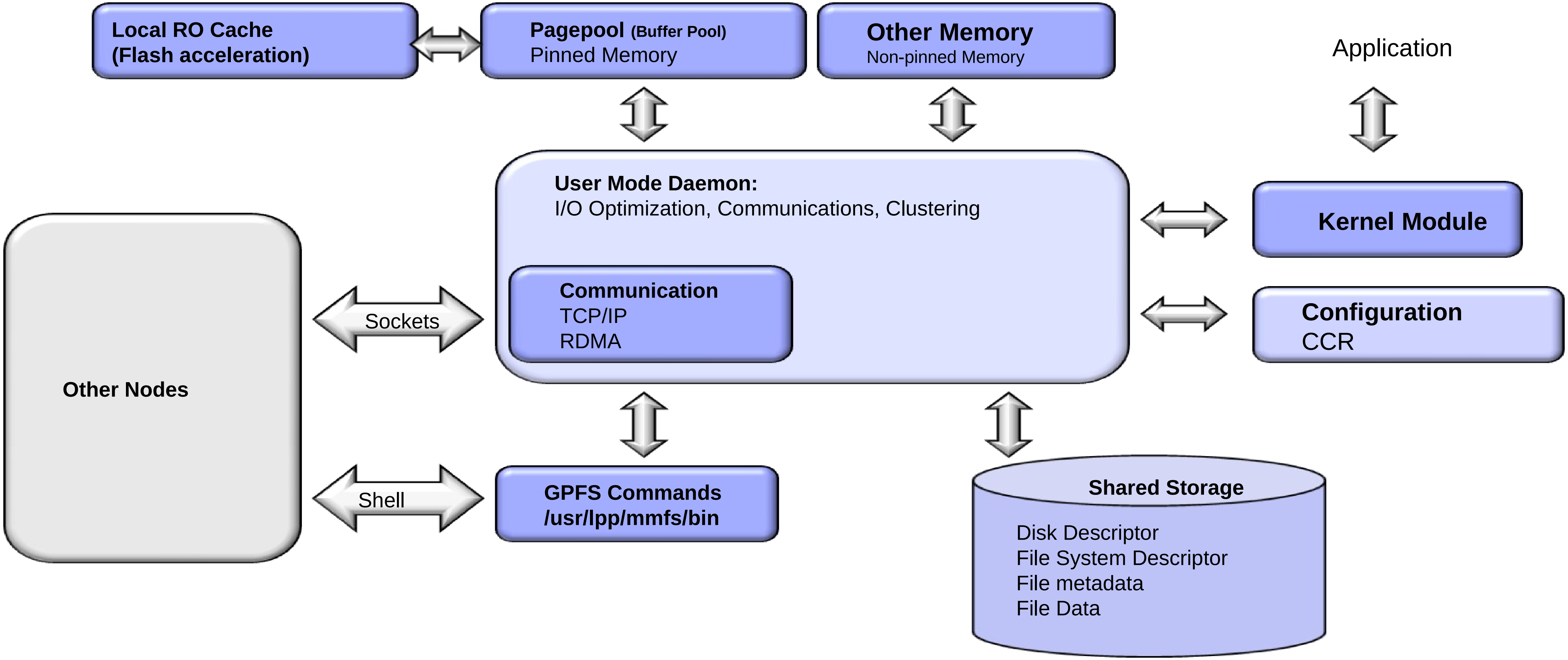
Token Manager/s

- Multiple per filesystem
- Each manage portion of tokens for each filesystem based on inode number

- There are other roles in a cluster: Gateways, helper nodes etc.

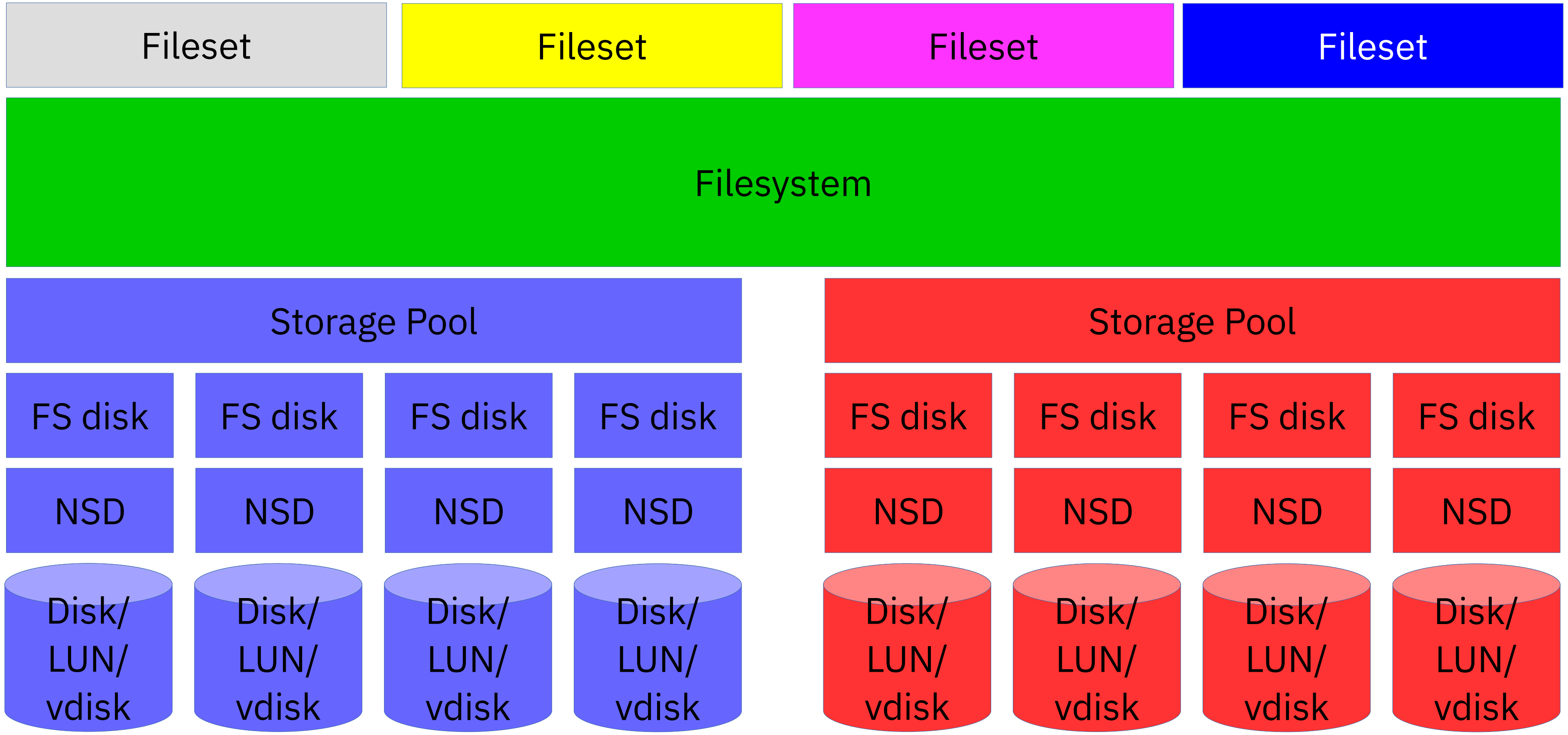
Node Internal Structure

- Spectrum Scale node based on a userspace daemon and kernel modules



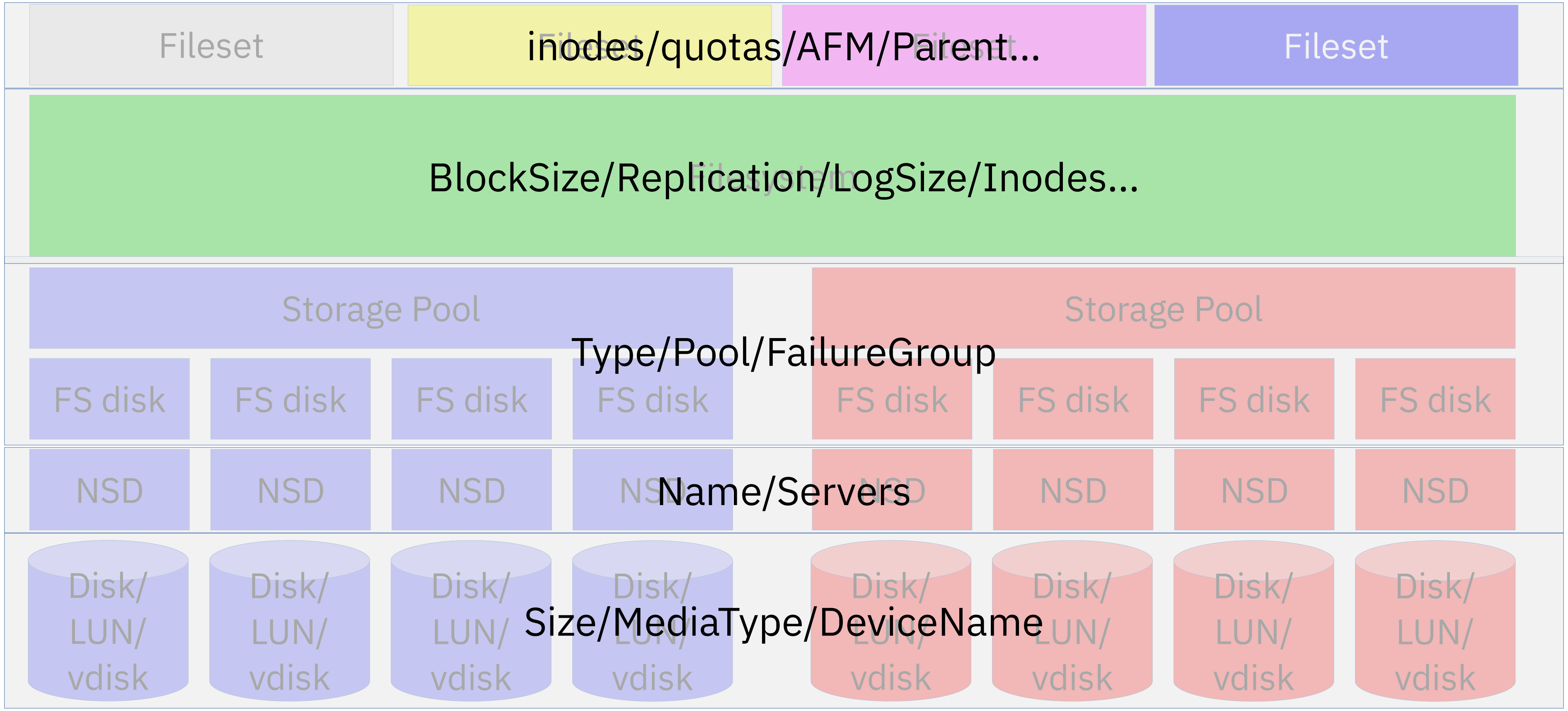
Filesystem Components

- Scale filesystem (GPFS) can be described as loosely coupled layers



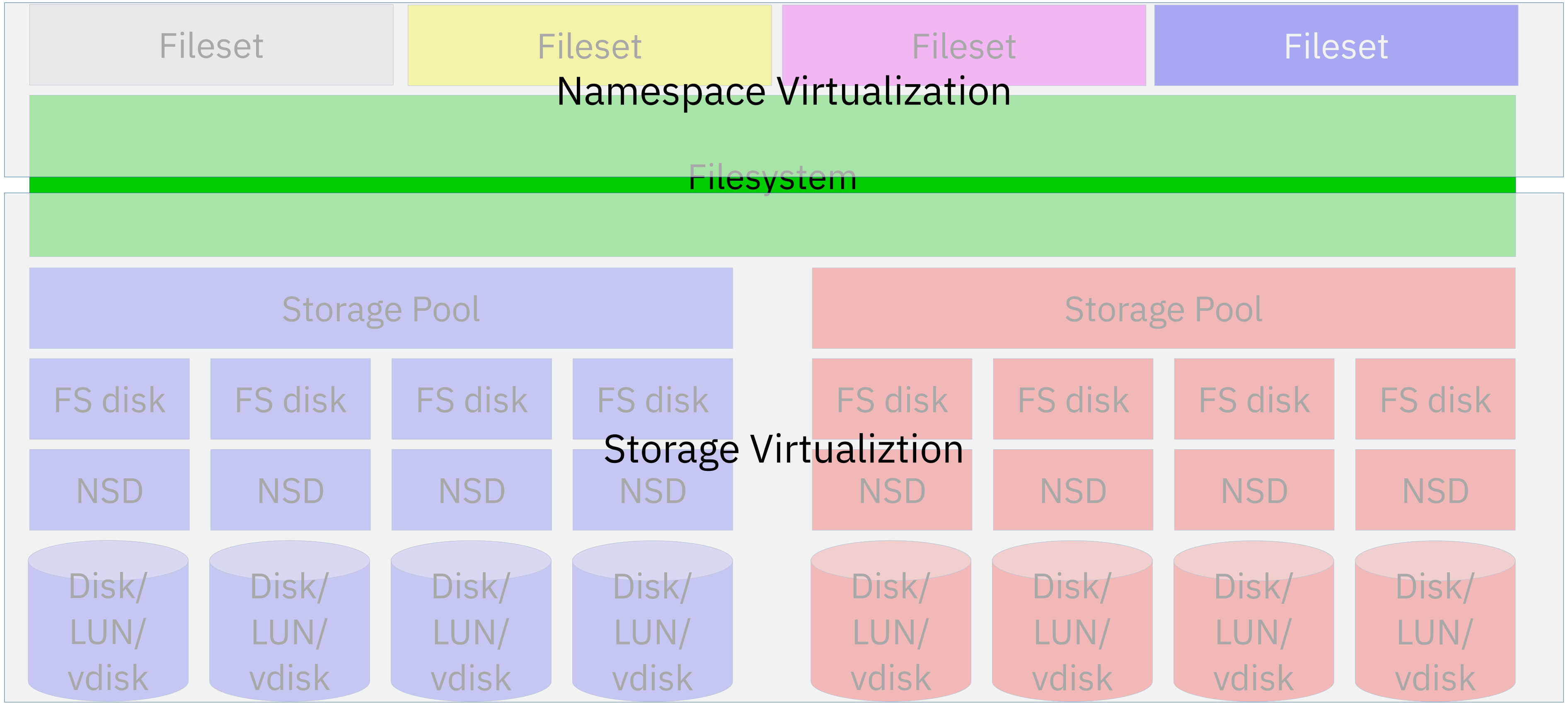
Filesystem Components

- Scale filesystem (GPFS) can be described as loosely coupled layers
- Different layers have different properties



Filesystem Components

- Scale filesystem (GPFS) can be described as loosely coupled layers
- Different layers have different properties



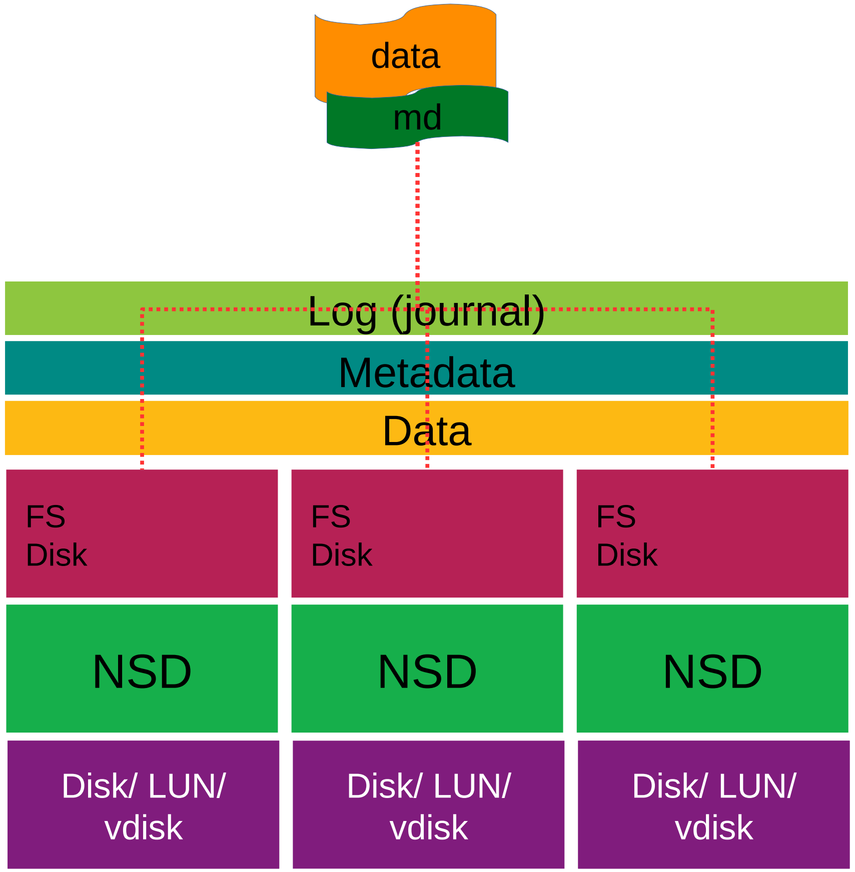
Filesystem Components

- Spectrum Scale stripes the data/metadata/journal between pooled disks in order to achieve scalable performance

- File data and metadata is striped on all relevant disks (depends on fs disk designation)

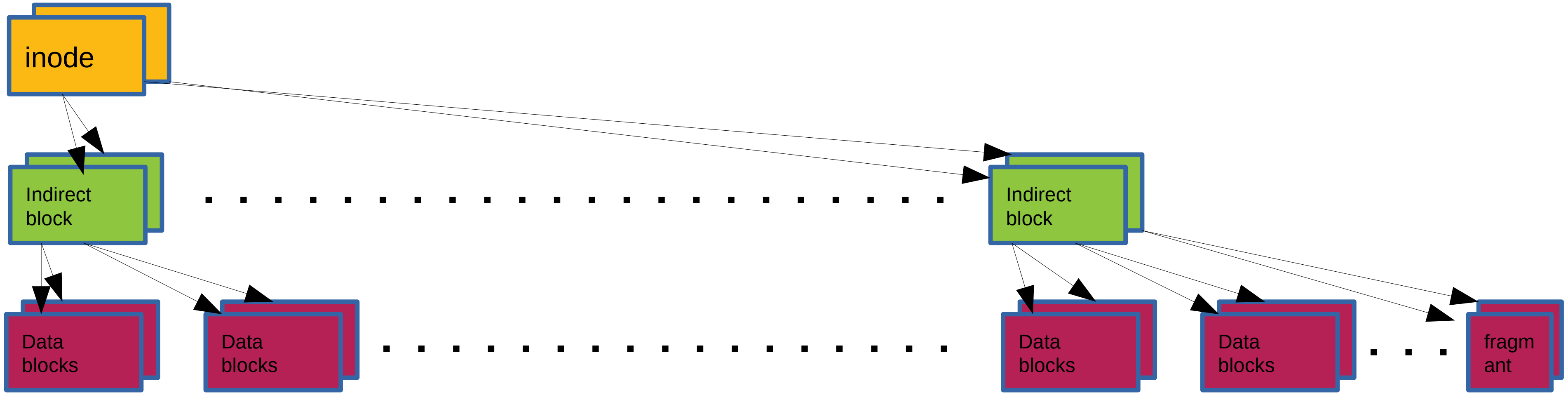
- Metadata, and potentially small writes are first committed to the FS journal (log) in order to be able to recover from “short writes”

- Data/metadata might flow using direct path and/or network depending on NSD configuration



Advanced functionality - Replication

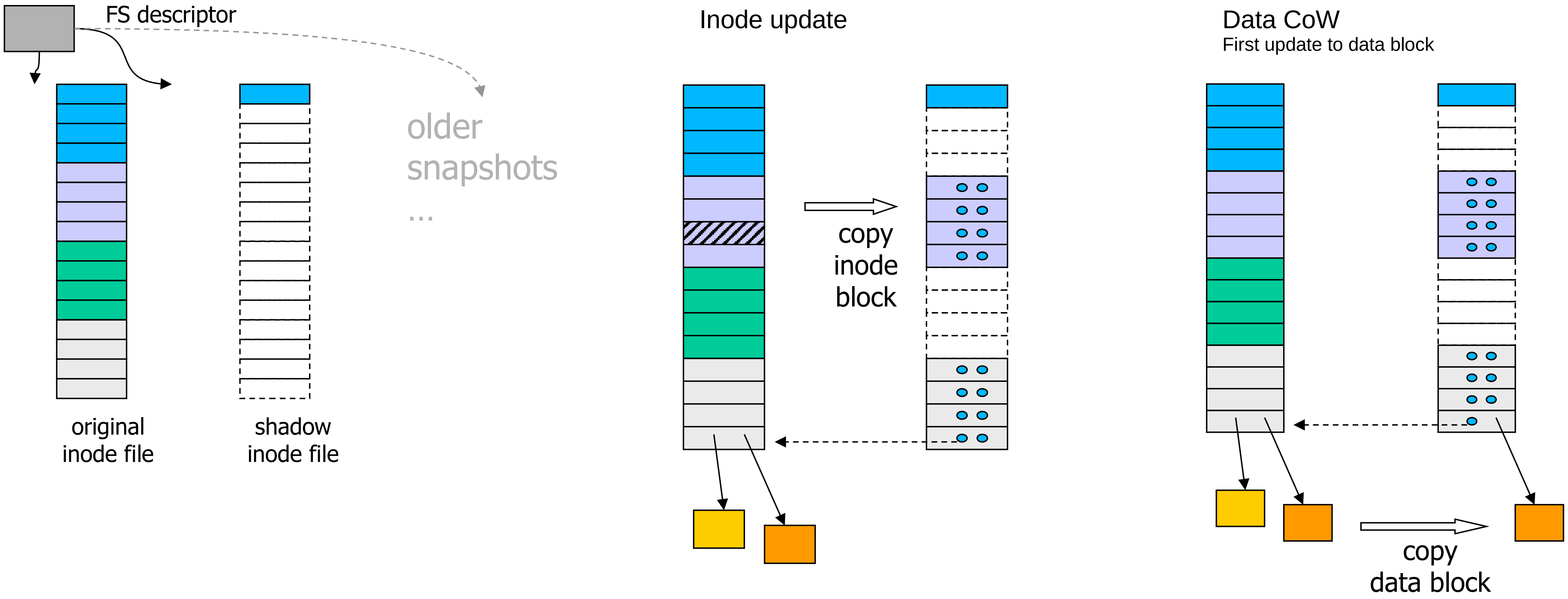
- Inode, indirect block and/or data blocks may be replicated
- Each disk address: list of pointers to replicas
- Each pointer: disk id + sector no.



- No designated “mirror”, each replica is stored in a different FG
- On read, using either round-robin, local or fastest (*)
- Ill-replication is recorded and fixed on a per indirect block basis (rapid repair)

Advanced functionality - Snapshots

- Snapshot: logical read-only copy of the filesystem at a point in time
- Snapshot data is accessible through the .snapshots directories
- Implemented using "shadow inode file"



FS descriptor

original inode file

shadow inode file

older snapshots …

Inode update

copy inode block

Data CoW

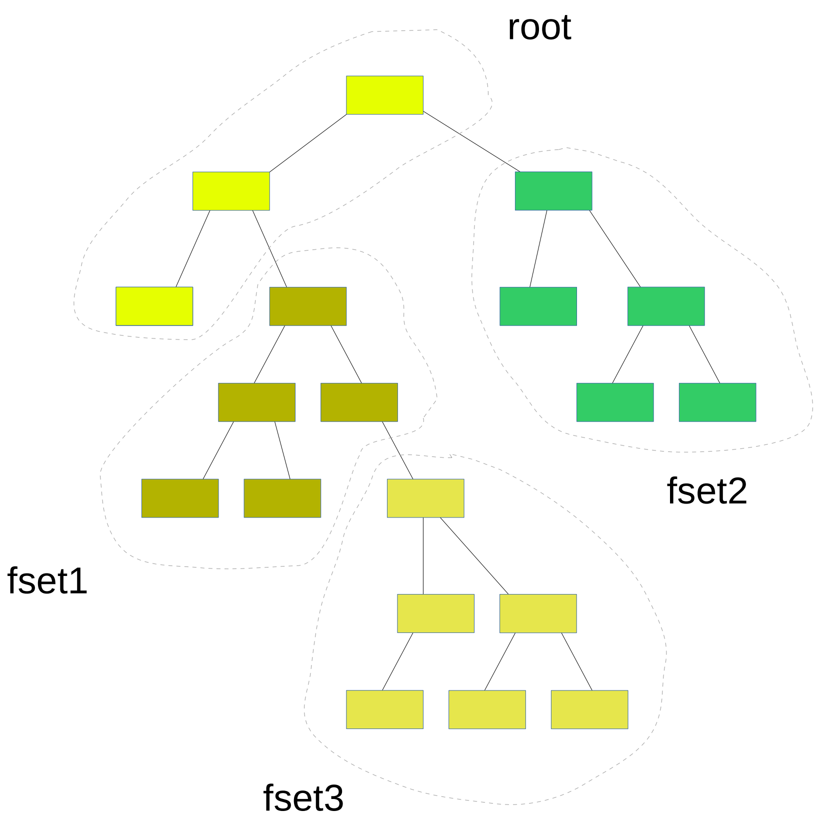
First update to data block

copy data block

Advanced functionality - Filesets

- “namespace virtualization”
- Fileset: A partition of the file system name space (sub-directory tree)
  - Allows administrative operations at finer granularity than entire file system, e.g.,

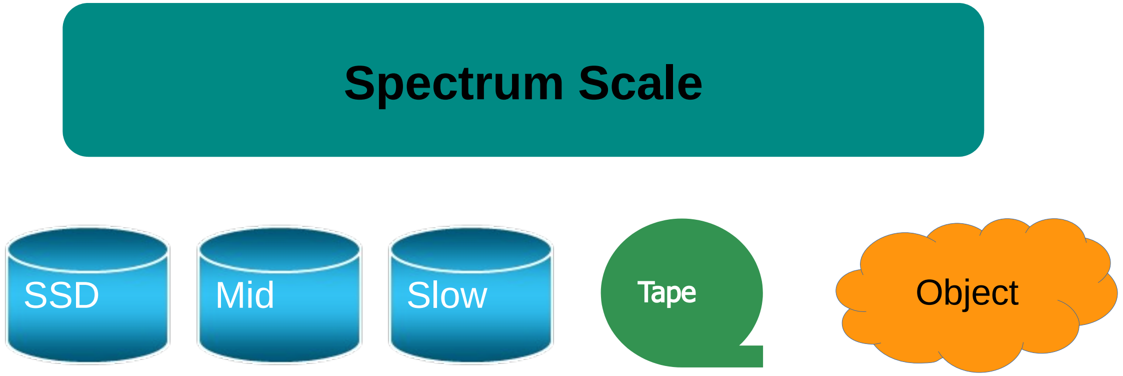
    disk space limits, user/group quota, snapshots, caching, ...
  - Can be used to refer to a collection of files in policy rules
- Independent fileset: A fileset with a reserved set of inode block ranges (“inode space”)
  - Allows per-fileset inode scan
  - Enables fileset snapshots (inode copy-on-write operates on inode blocks)
  - Separate inode limit and inode file expansion for each inode space
    → Active inode file may become sparse



Advanced functionality – ILM/HSM

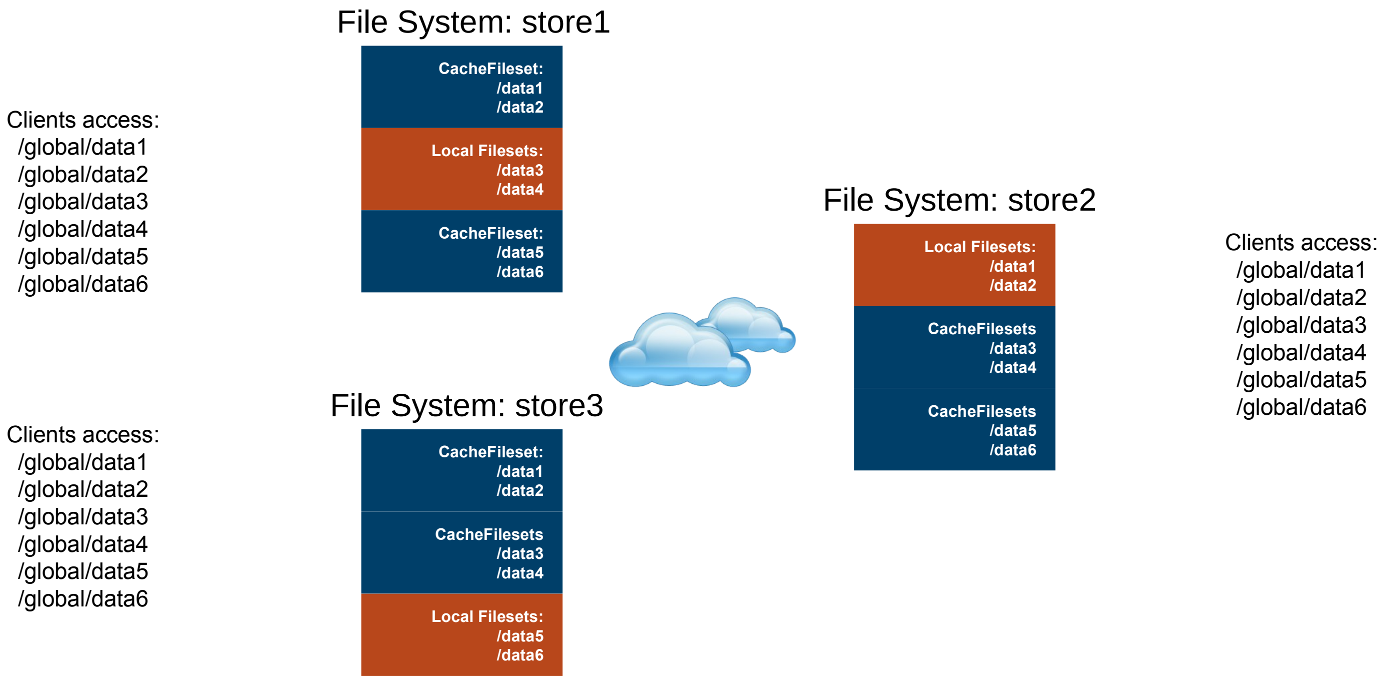
- “Store the data on the right media”
- Using policy and/or fast inode scan in order to make decisions (placement, migration)
- Block placement is managed by the filesystem in any case – so its transparent to users
- External storage pools (using either DMAPI or LWE) can be used as a “cold” tier

```
RULE 'DefineTiers' GROUP POOL 'TIERS'
            IS 'gold' LIMIT(80)
            THEN 'silver' LIMIT(90)
            THEN 'bronze'
RULE 'Rebalance' MIGRATE FROM POOL 'TIERS' TO POOL
'TIERS' WEIGHT(FILE_HEAT)
```



Advanced functionality – AFM

- Asynchronous file caching with “side effects” (DR)
- Creating a global distributed namespace
- Currently uses either NFS or native NSD protocol (pros and cons)
- Logic is usually at the cache only – Single gateway coordinates, all gateways can do the data movement



File System: store1

CacheFileset:
/data1
/data2

Local Filesets:
/data3
/data4

CacheFileset:
/data5
/data6

Clients access:
/global/data1
/global/data2
/global/data3
/global/data4
/global/data5
/global/data6

File System: store2

Local Filesets:
/data1
/data2

CacheFilesets
/data3
/data4

CacheFilesets
/data5
/data6

Clients access:
/global/data1
/global/data2
/global/data3
/global/data4
/global/data5
/global/data6

File System: store3

CacheFileset:
/data1
/data2

CacheFilesets
/data3
/data4

Local Filesets:
/data5
/data6

Clients access:
/global/data1
/global/data2
/global/data3
/global/data4
/global/data5
/global/data6

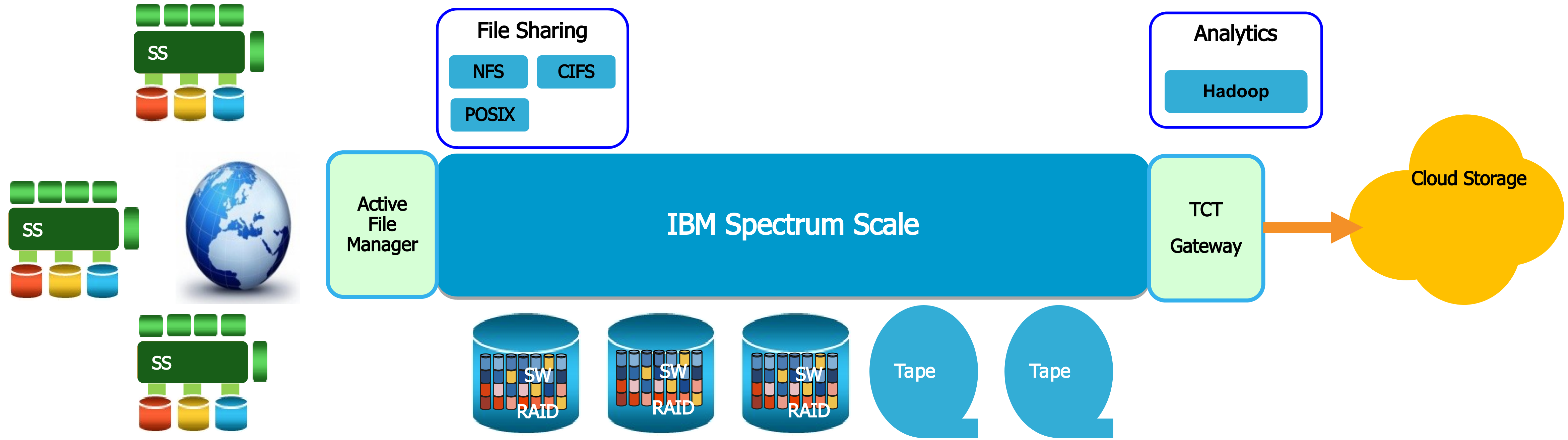
Advanced functionality – Encryption/compression

- **Encryption**
  - Encryption (as everything else) takes place at the client side
  - **Data** travels encrypted over the network (but not metadata) – tscomm cipher may be required (at rest vs. in transit)
  - Keys are stored on a key server
  - Per node key granularity (multi-tenancy?)
  - Managed by policy engine (different/multiple keys per filesystem object)
  - Allows secure deletion (but might be expensive)
- **Compression**
  - Compression (as everything else) takes place at the client side
  - Improved network and caching (due to compression)
  - Expandable algorithm
  - Managed by policy engine

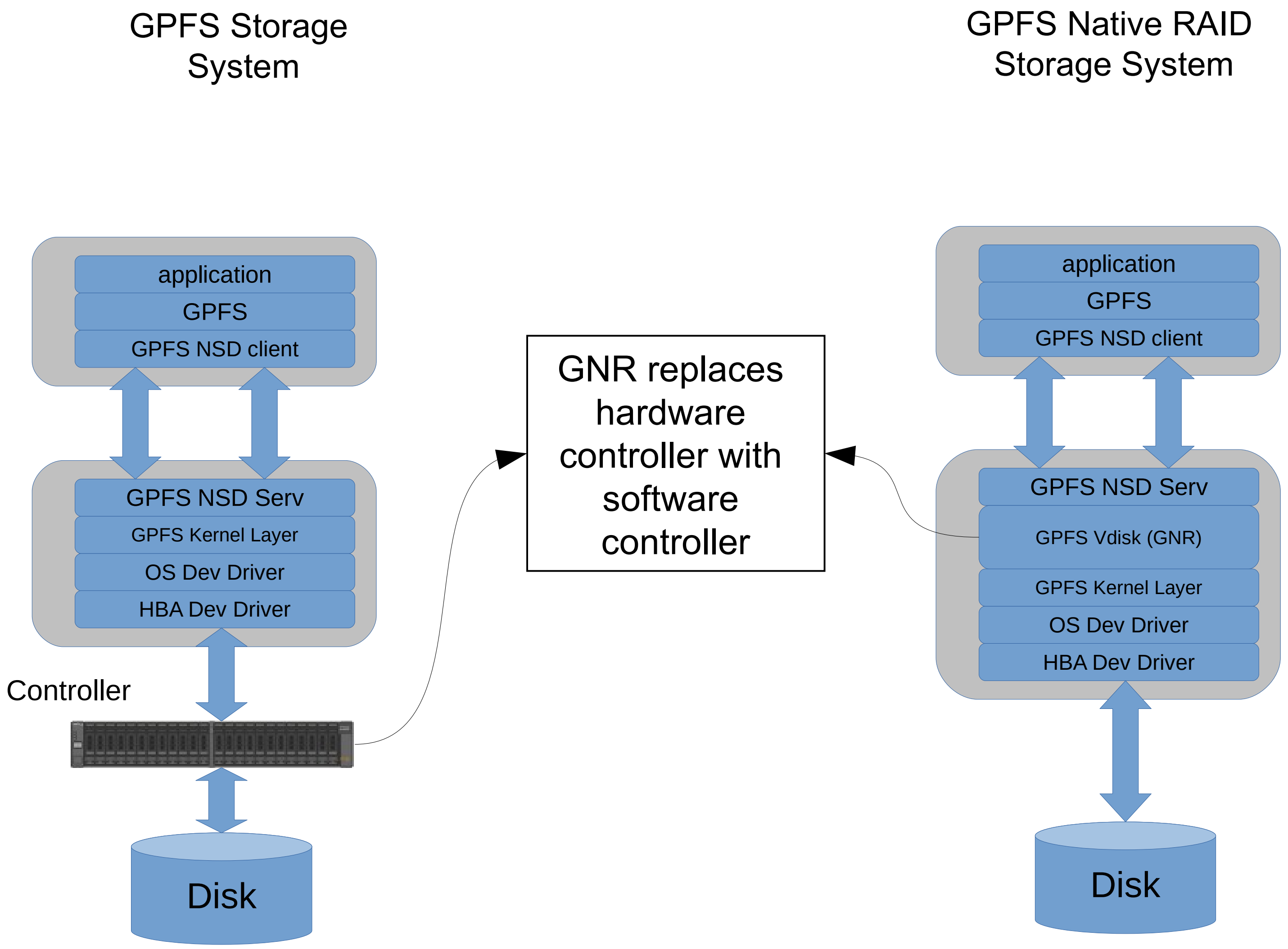
Adding declustered RAID

From Clustered Filesystem into Universal Storage Platform

- Reduce space overhead of replication using declustered parity. (80% utilization vs 33%)
- Extremely fast rebuild on failure with much less performance impact
- Packaged as hardware solution (ESS/DSS) or software (ECE)



- **Spectrum scale SW raid: getting closer to the data**

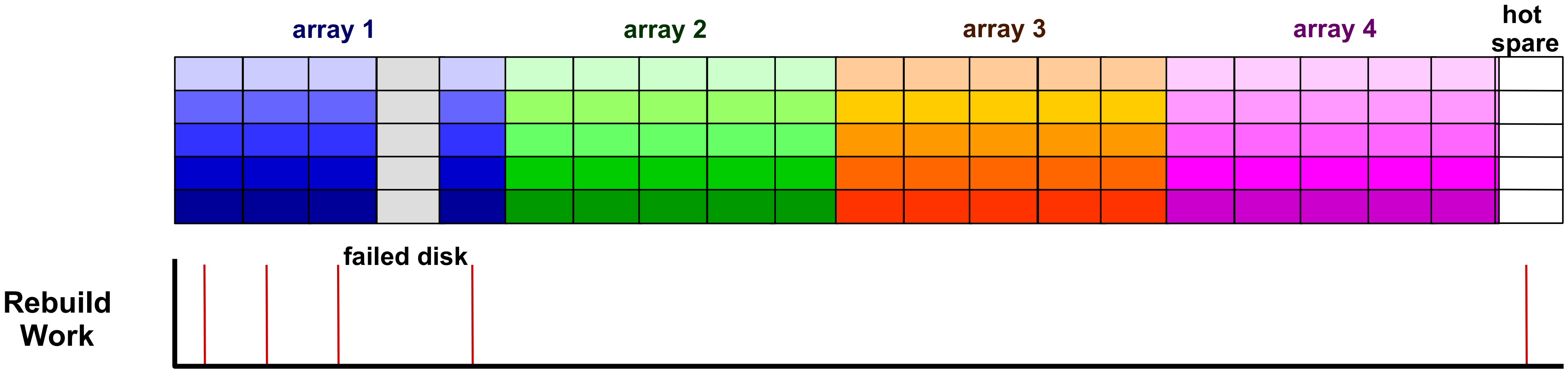


- Embedded within Network Shared Disk (NSD) layer of GPFS
- Utilizes generic servers with direct-attach SBOD disks
- OS: Linux/Power, Linux/x86, AIX/Power, and potentially others.
- Scalable from small systems to large supercomputers (10 - 100,000 disks)
- Developed under the DARPA PERCS program

GNR – Conventional RAID explained

Conventional RAID Limits RAID Rebuild Performance

- RAID rebuild operations span a limited number of disks because a RAID group maps to a small subset of pdisks. This provides fewer disks for RAID rebuild operations.
- Example: 21 disks @ 4 x 3+P+Q RAID-6 arrays with 1 hot spare



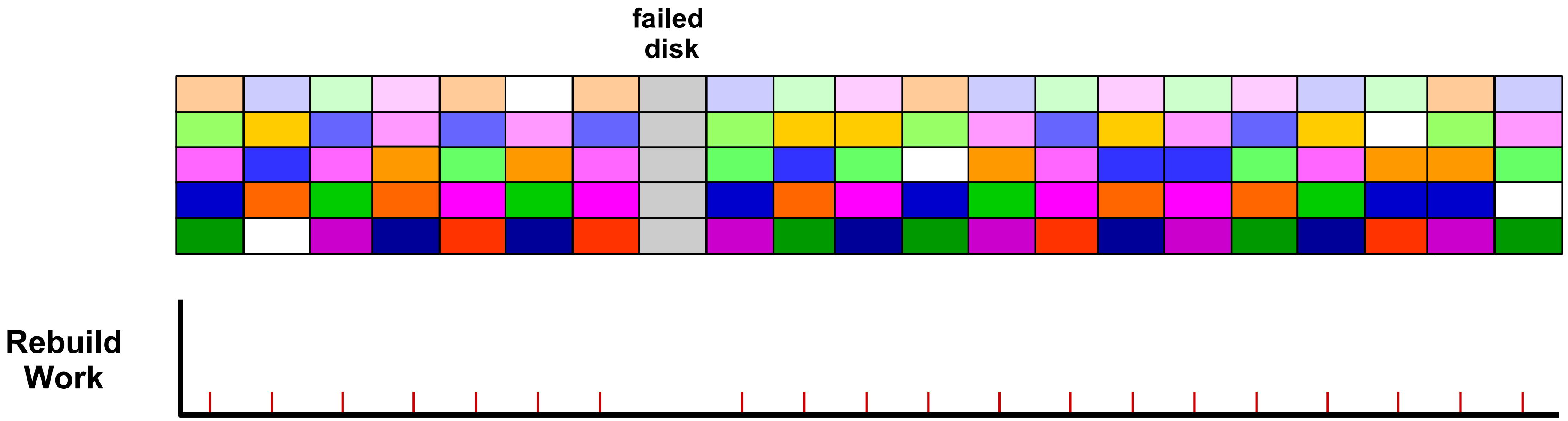
Only 5 disks participate in the rebuild operation - the 4 good disks and the hot spare.

GNR - Declustered RAID explained

Declustered Arrays Improve RAID Rebuild Performance:

- RAID rebuild operations span all disks in the DA because the RAID groups span all disks in the DA. This provides more disks for RAID rebuild operations.
- Example: 21 disks configured with 1 virtual spare disk

Data from any virtual 3+P+Q vdisk is distributed across all pdisks in DA

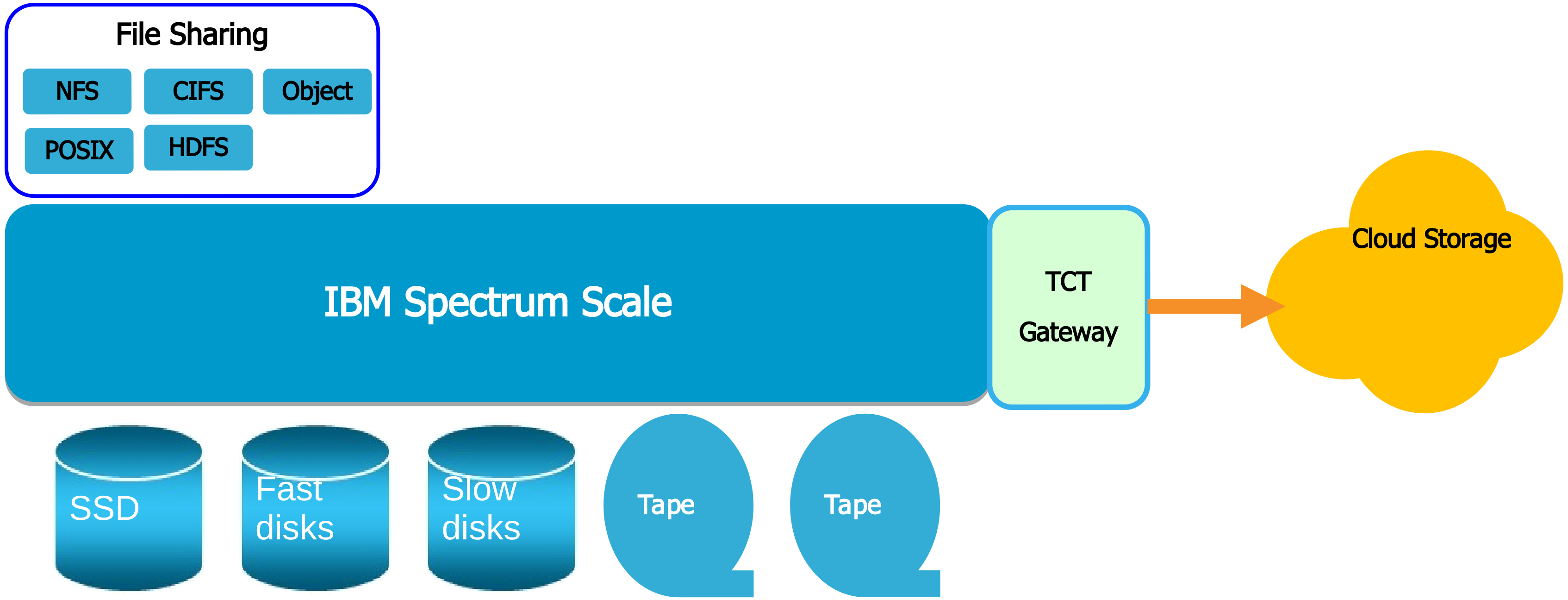


All 20 surviving disks participate in the rebuild.

Adding NAS capabilities

From Clustered Filesystem into Universal Storage Platform

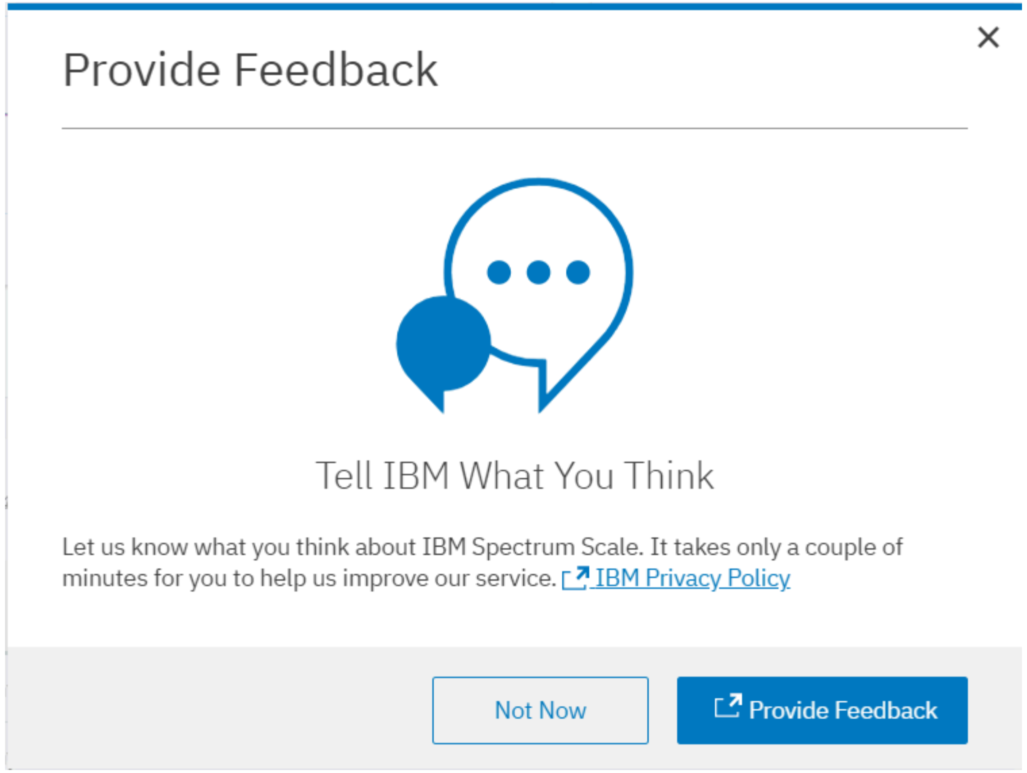
- NFS Support was added in Spectrum Scale 3.2, CIFS since 3.4 on SONAS/V7000 Unified
- CIFS Support was first released to the market as part of SONAS and V7000U only and ships as a pure software feature as part of Spectrum Scale starting in version 4.1.1 (TL2)
- HDFS and Object (S3/Swift) was integrated into the product as well



Questions ?

TWENTY QUESTIONS
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
DICK
HERB
VAN
FLORENCE
ALDO RAY

Thank You



Provide Feedback

Tell IBM What You Think

Let us know what you think about IBM Spectrum Scale. It takes only a couple of minutes for you to help us improve our service. IBM Privacy Policy

Please help us to improve Spectrum Scale with your feedback

- If you get a survey in email or a popup from the GUI, please respond
- We read every single reply