

AFM Deep Dive

A person wearing glasses and a dark shirt is standing in a server room, looking at a laptop. The room is filled with server racks, and the lighting is dim with some blue and green lights visible on the racks. The perspective is looking down a long aisle of server racks.

IBM Storage Scale Days 2024

March 5-7, 2024 | Stuttgart Marriott Hotel Sindelfingen

Venkat Puvvada (vpuvvada@in.ibm.com)

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

IBM reserves the right to change product specifications and offerings at any time without notice. This publication could include technical inaccuracies or typographical errors. References herein to IBM products and services do not imply that IBM intends to make them available in all countries.

IBM's Global Data Platform for File & Object Data



1 Data Access Services



2 Data Caching Services

Global Data Platform (powered by Spectrum Scale)



Investment protection

File & Object Storage
(NetApp, PowerScale, etc)

Object Storage

IBM COS

File Storage

Spectrum Scale

NextGen workloads





Spectrum Fusion

3 Data Management Services

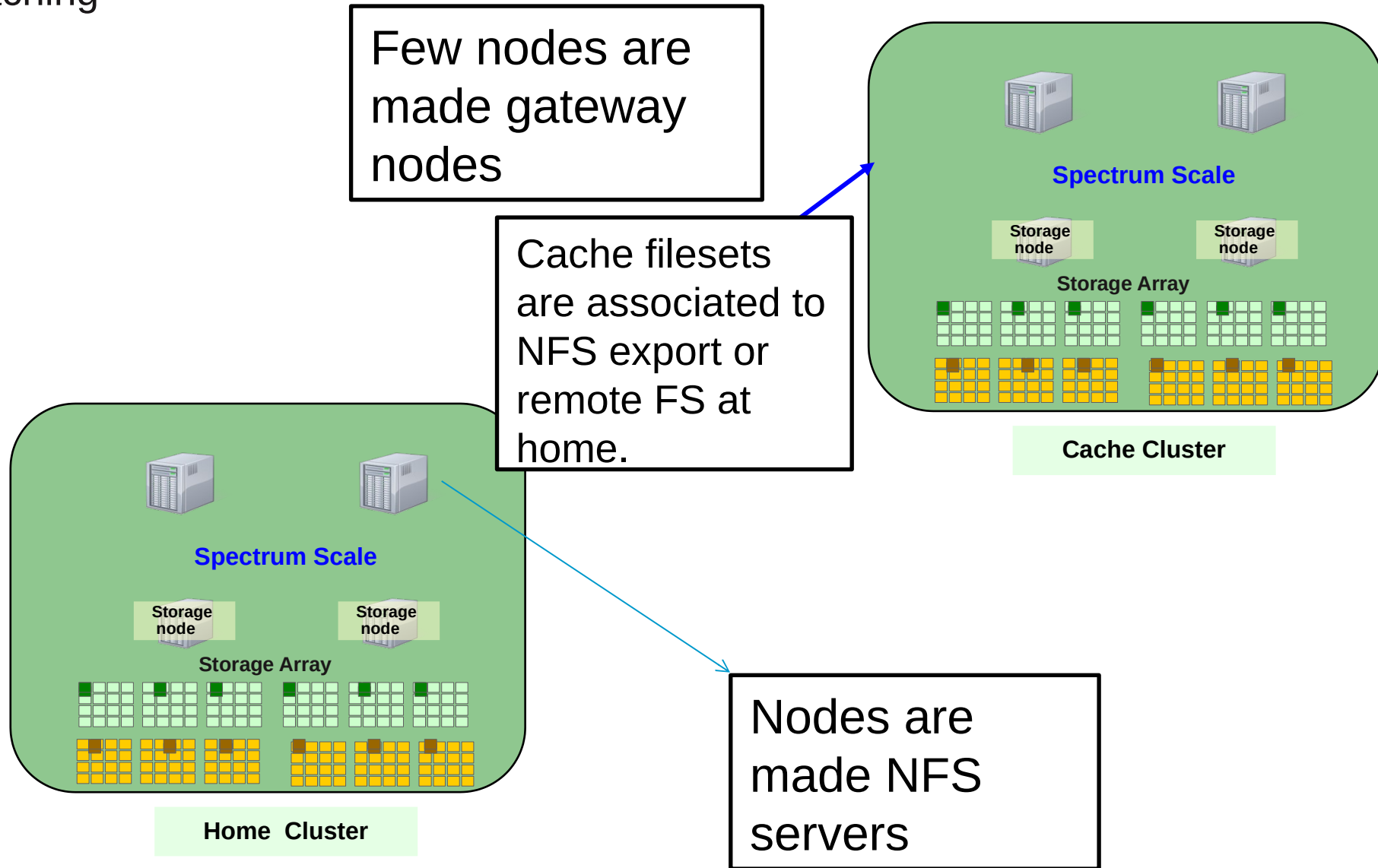
4 Data Security Services

- Identify
- Protect
- Detect
- Respond
- Recover

Data Caching Services (Active File Management) Use cases

			
<h3>Data Virtualization</h3>	<h3>Data Collaboration</h3>	<h3>Data Resilience</h3>	<h3>Hybrid cloud / Bursting</h3>
<ul style="list-style-type: none"> • Integrate legacy file and object data stores into single file system to breakdown legacy data silos • Migrate data to new storage or continue to use legacy stores • Create a High-Performance Tier for analytics for legacy data with transparent data access 	<ul style="list-style-type: none"> • Geo-distributed collaboration on data transparently shared between data centers, the cloud and edge sites • Coalesce data to a home site from the edge and redistribute it to all sites 	<ul style="list-style-type: none"> • Provide an asynchronous Disaster Recovery solution for business continuity over WAN distances • Supports analytics and archival access to passive data 	<ul style="list-style-type: none"> • Dynamically increase computation resources in the cloud and optimally make required data available for Cloud bursting • Process data consolidated on S3 Cloud Storage on with high performance tier in the Cloud Compute Cluster • Archive data to S3 Object storage
<p>Public Cloud Services</p> <p>Use case:</p> <p>Enables end user service to upload large amount of data via Object interface that can be analysed on high performance file system</p>	<p>Research / University</p> <p>Use case:</p> <p>Generate 100's of TB per day across multiple silos, leveraged AFM to provide common namespace with transparent multiprotocol data access</p>	<p>Multinational financial services</p> <p>Use case:</p> <p>Disaster Recovery, retention and compliance data with FileNet and ESS</p>	<p>Research Biopharmaceutical</p> <p>Use case:</p> <p>Multi site / public cloud bursting for collaboration</p>

AFM WAN Caching



Few nodes are made gateway nodes

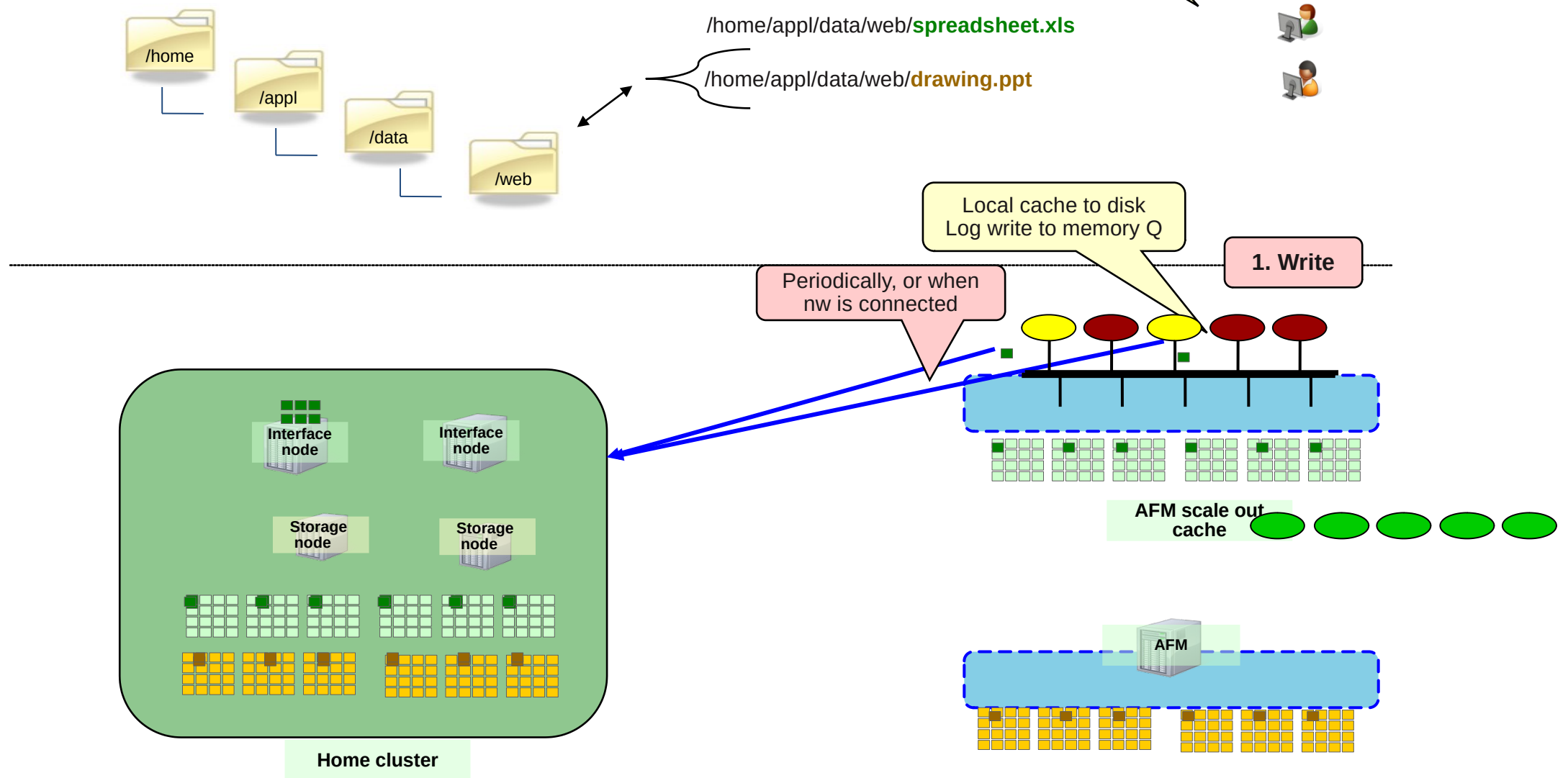
Cache filesets are associated to NFS export or remote FS at home.

Nodes are made NFS servers

Definitions

- Node types
 - **Application node**
 - Writes/reads data based on application request to the GPFS filesystem at cache cluster
 - Can be Linux/AIX/Windows – NFS/SMB mounts
 - **Gateway node(s)** is the node that connects to the home cluster
 - Queue of pending operations is in memory at the gw node
 - Selected based on hash of filesetid or user-defined.
 - Reads/writes data from the home cluster to the cache cluster
 - Checks connectivity with the home cluster and changes to disconnected mode on connection outage
 - Triggers recovery on failure
 - Only Linux supported
- Sites
 - Home cluster (Object Store)
 - Exports a fileset that can be cached
 - Cache cluster
 - Runs AFM and “connects” a local fileset with the home fileset.
- Transport Protocol
 - NFS, NSD and S3

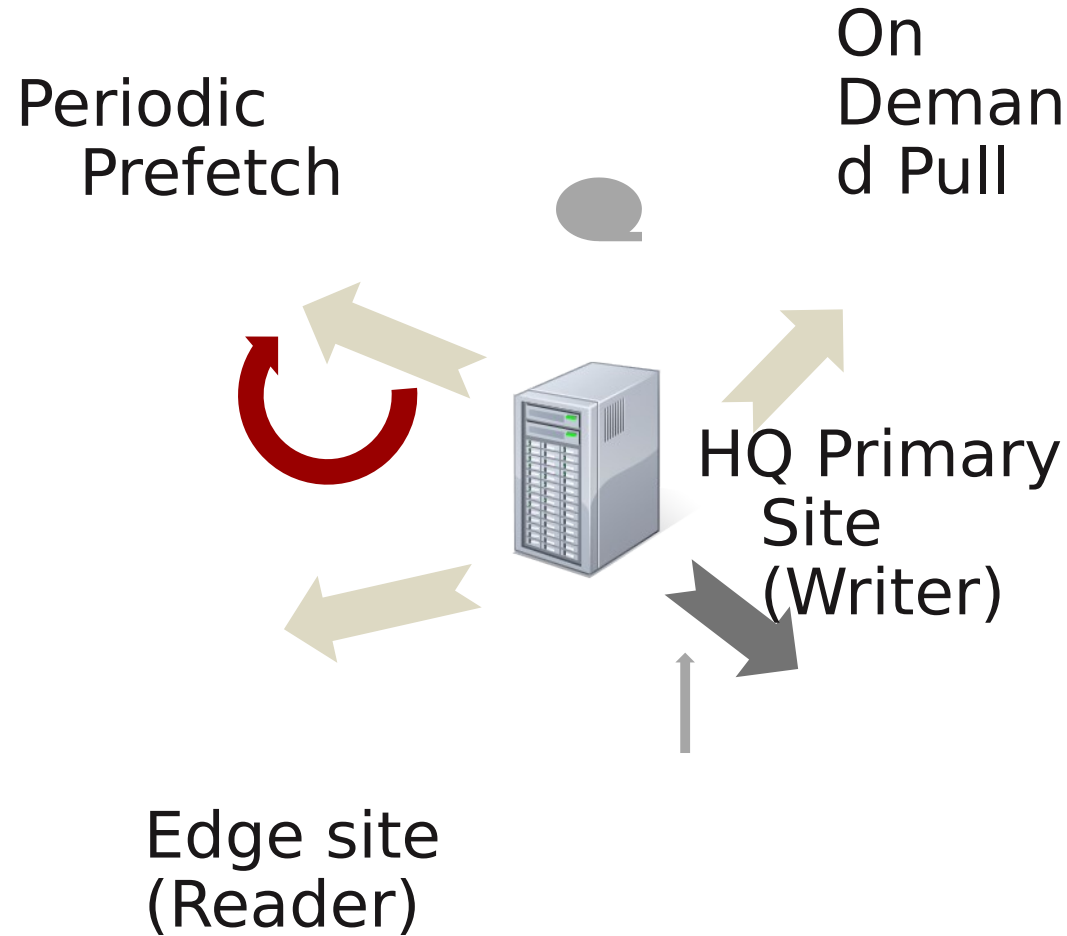
Asynchronous write back



AFM Modes

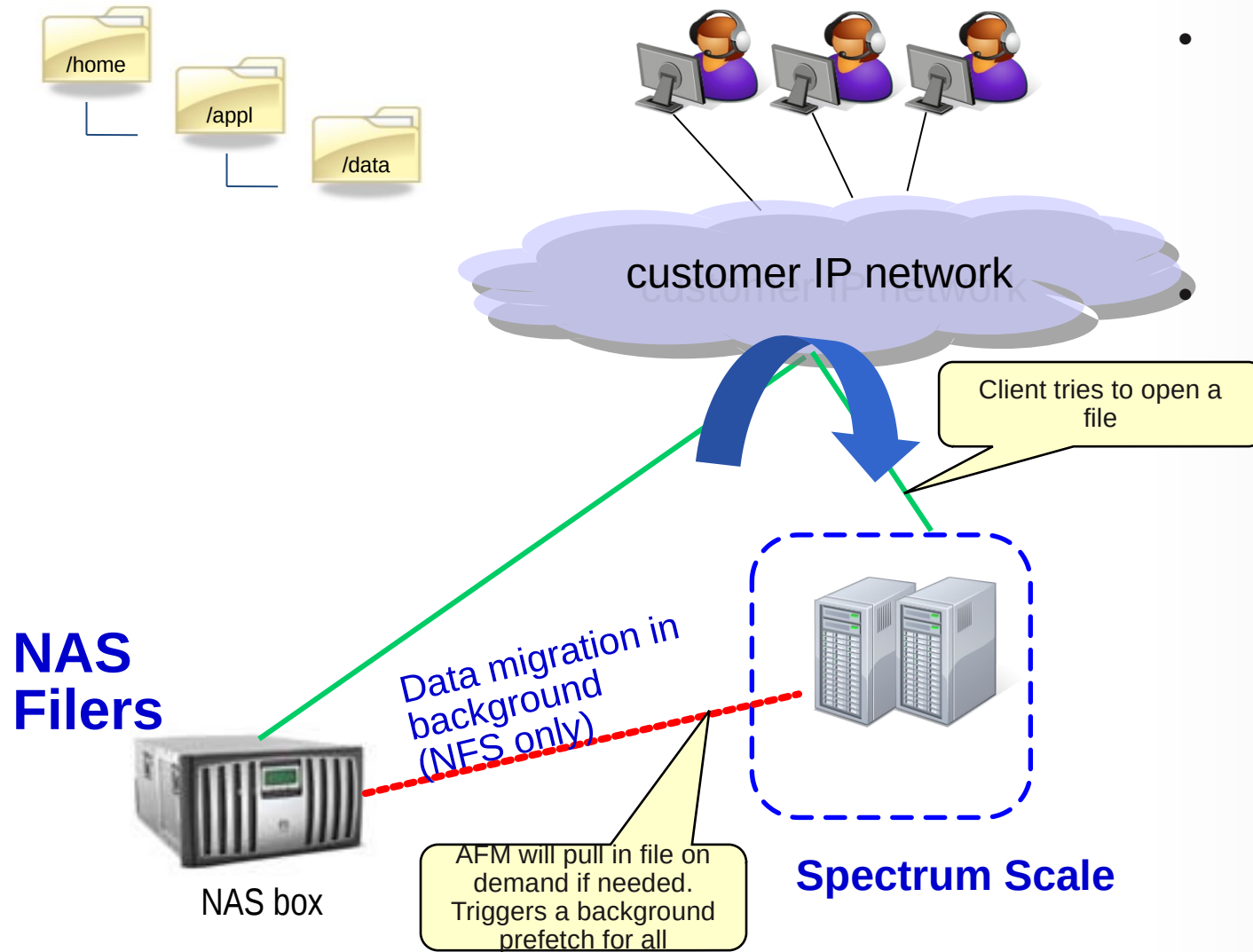
- **Single Writer**
 - Only cache can write data. Home can't change. Other peer caches have to be setup as read only caches.
- **Read Only**
 - Cache can only read data, no data change allowed.
- **Local Update**
 - Data is cached from home and changes are allowed like SW mode but changes are not pushed to home.
 - Once data is changed the relationship is broken i.e cache and home are no longer in sync for that file.
- **Independent Writer**
 - Allow Multiple Caches to be writers.
 - Each cache typically writes to independent files (no conflicts)
 - In case two caches write to same file, the last write that gets shipped to home wins. In essence last writer wins(data & metadata)
 - Data at home can change and revalidation happens.

Use Case: Content Distribution (Central/Branch Office)



- Central Site is where data is created, maintained, updated/changed.
- This is typically done in customer situations, where data is ingested via satellite or data warehousing etc.
- Branch/edge sites can periodically prefetch (via policy) or pull on demand
- Data is revalidated when accessed
- A typical scenario for this is:
 - Music sites, where data is maintained at central location and other sites in various locations will pull in data into cache and serve the data locally at that location.

(Legacy) NAS Migration

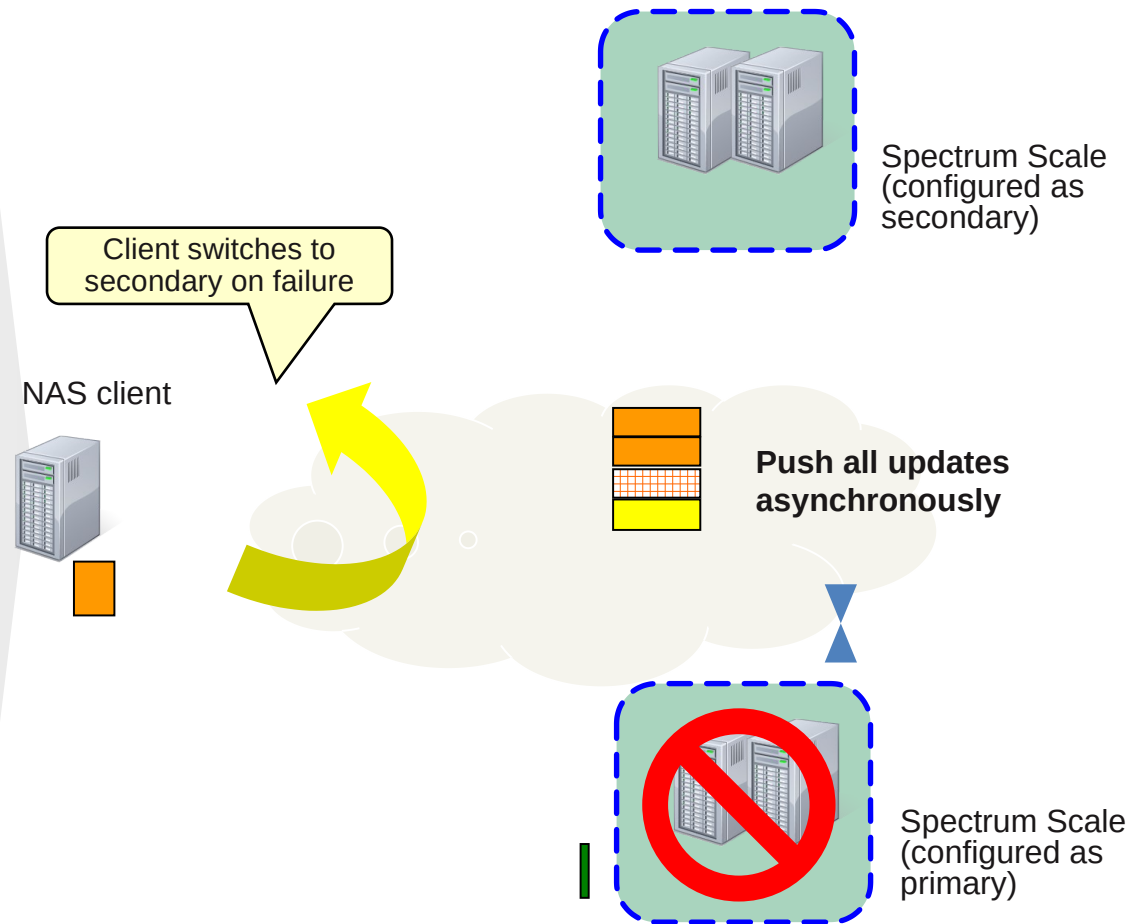


- **Clients switched to new Spectrum Scale box immediately**
- **Data movement in background**

- AFM will fetch data on read miss
- Trigger background prefetch
- All updates done to AFM will commit locally

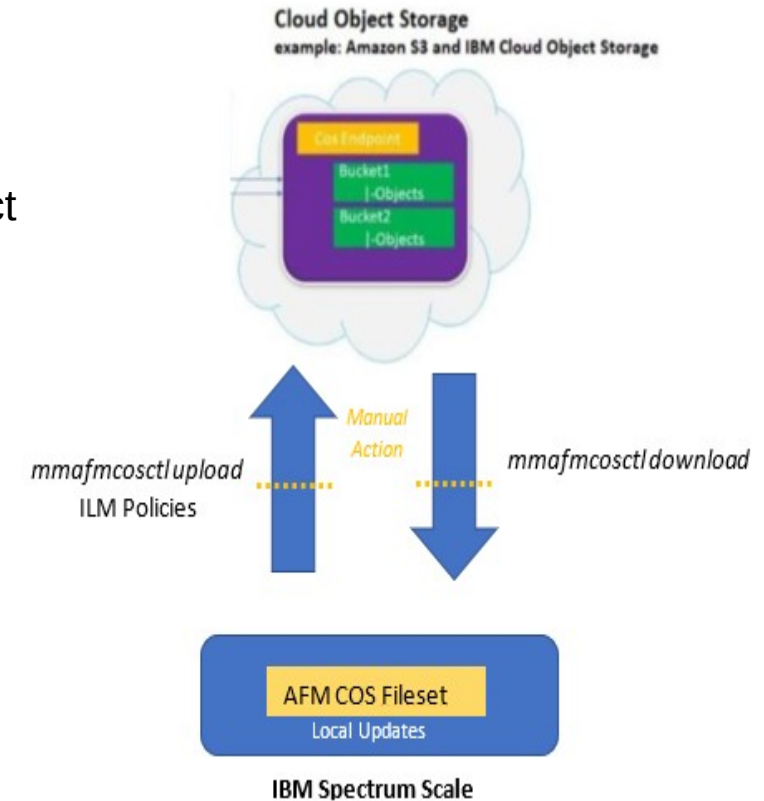
Disaster Recovery with AFM Async Replication

- Establish per-fileset replication relationship using AFM
 - Configure AFM in a primary-secondary relation
- Only deltas are pushed
 - AFM tracks exact filesystem operation
 - Only updated byte range will be pushed
- Multi-site Snapshots for consistent copy
 - Snapshots at both sides are in sync.

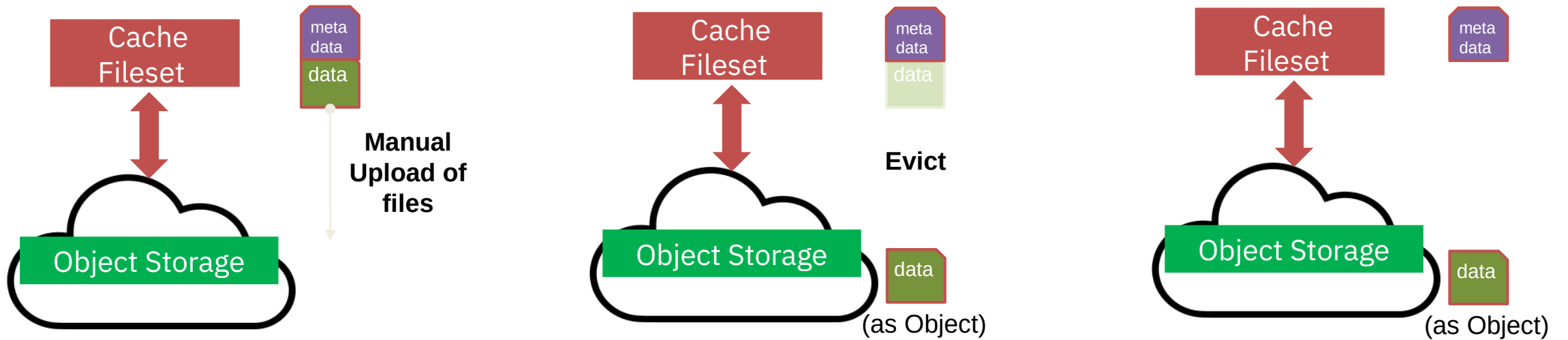


AFM for archive use case with Manual Update (mu) mode

- Supports manual upload/ download of objects using ILM policies or object list and avoids automatic upload/ download
- All changes are local. Manual action is required to upload or download files to cloud object storage
- Metadata is refreshed only once when the MU fileset is created pointing to non-empty bucket
- Manual control over deletion from cache
- Manual deletion from Cloud Object Storage
- An independent fileset can be converted to a MU mode AFM fileset
- Auto removal of files/ objects from Cache and Object Storage using fileset parameter 'afmMuAutoRemove'
- Specify policy to be used with `mmafmcosctl reconcile -policy` command



AFM for archive use case with Manual Update (mu) mode



Recent AFM S3 Object storage features

- Support of STS token for AFM to cloud object fileset
- Support for expiring and automatically refreshing cloud storage keys
- Tunable to manage key expiration using 'afmObjKeyExpiration'
- Support of creating and upload objects for empty directories in AFM to cloud object storage.
- Support of marking files and directories as local in AFM to cloud object storage fileset.
 - `#mmafmctl fs setlocal -j AFMtoCOS --path /ibm0/fs/AFMtoCOS/file1`
- Support of adding user defined prefix in AFM to cloud object storage fileset.
 - `#mmafmcconfig fs1 afmbktprefix1 --endpoint https://region@endpoint --object-fs \
--xattr--prefix dir1 --bucket bkt1 --acls--mode sw`
- Support of replicating more than 2K metadata in AFM to cloud object storage fileset.
- Support for outband download of objects.

AFM Cloud Object Storage Operation modes

ObjectFS mode

- Behaves like normal AFM modes fileset.
- Objects are downloaded on read or on access
- IW and SW modes push files to cloud object storage
- RO, LU, IW automatically pulls objects from the cloud object storage and stores as files.

ObjectOnly mode

- Default for object operation mode
- No on-demand refresh on read
- Need to manually download metadata/data from COS.
- Objects are uploaded automatically (IW and SW)
- Avoids frequent trips and reduce network contention by selective download/uploads.

	ObjectFS	ObjectOnly
Read Only (RO)	Upload - NA Download - On access (Auto)	Upload - NA Download - On demand
Local Update (LU)	Upload - NA (only On demand) Download - On access (Auto)	Upload - NA (only On demand) Download - On demand
Single Writer (SW)	Upload - Auto Download - On access / On demand	Upload - Auto Download - On demand
Independent Writer (IW)	Upload - Auto Download - On access (Auto)	Upload - Auto Download - On demand
Manual Update (MU)	Upload - On demand Download - On demand	

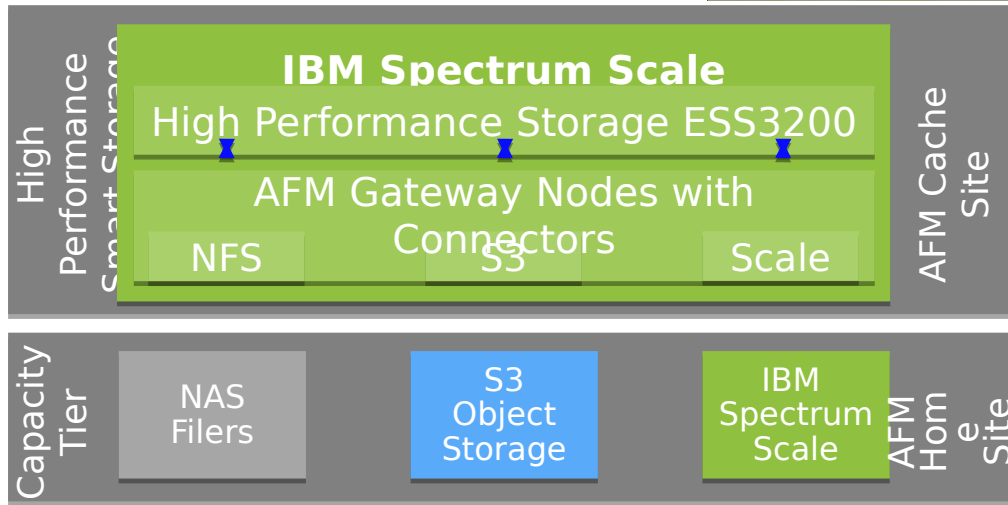
Important AFM S3 Object configuration parameters

- **afmMUAutoRemove**
 - Enabling this for a MU fileset will queue file remove operations to the COS automatically
- **afmObjectDirectoryObj**
 - Enabled at filesset level to support synchronisation of all directories with/without object with COS
- **afmObjectFastReaddir**
 - Enabled at the filesset level to skip the fetching of extended attributes and acl from COS for readdir operations leading to faster listing operations
- **afmObjKeyExpiration**
 - Specifies the COS key expiration timeout value which allows to reload the access/secret keys after the defined timeout value. Default is 36000 seconds. Set at the cluster level
- **afmParallelReadThreshold**
 - Defines the threshold beyond which parallel reads become effective and is enabled by default . Default is 1024MB.
- **afmParallelReadChunkSize**
 - Defines the minimum chunk size of the read that needs to be distributed among the gateway nodes during parallel read. Default is 128MB . A value of zero disables parallel reads
- **afmPrefetchThreshold**
 - Controls partial file caching. Default is 0 which caches the entire file by pulling all the blocks when 3 blocks are read at the cache. Values are in the range 1-100 and specified the percentage of file that must be read to cache the whole file. A value of 100 disables full file prefetching.
- **afmAsyncDelay**

Time by which the asynchronous operations are delayed to home

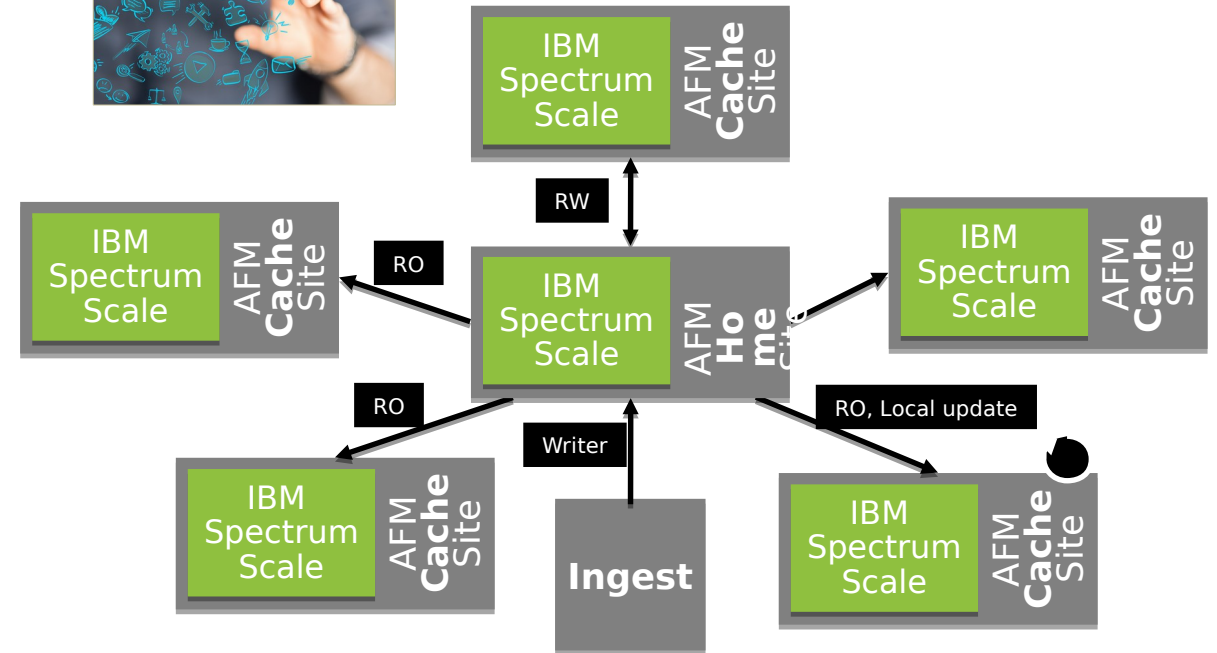
Data Caching Services (AFM) Use case details

Data Virtualization



- Vertical caching
- Common namespace across isolated data silos in legacy 3rd party data stores
- Transparent access to all data regardless of silos
- Scale-out Posix performance
- Data export via NFS, SMB, HDFS, Object
- Can be used to seamlessly migrate data to new storage

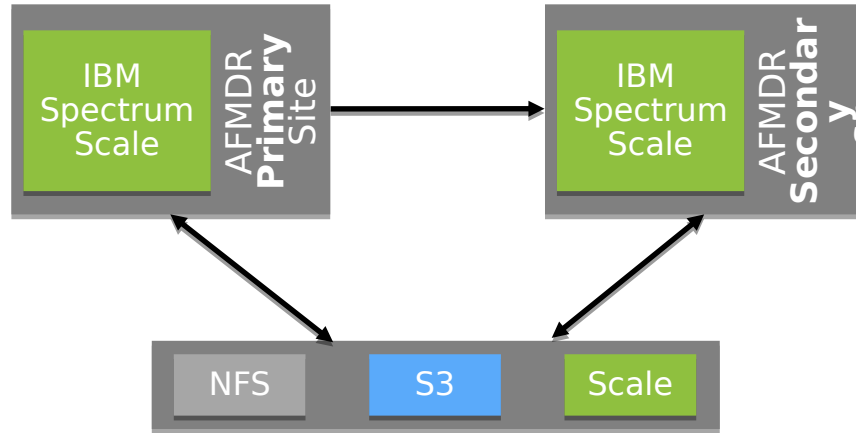
Data Collaboration



- Consistent cache provides a single source of truth with no stale data copies
- Horizontal caching
- Bi-direction traffic from Edge to Center
- Eventually Consistent data cache
- Transparent on-demand data access and transfer
- Policy driven data prefetch and eviction

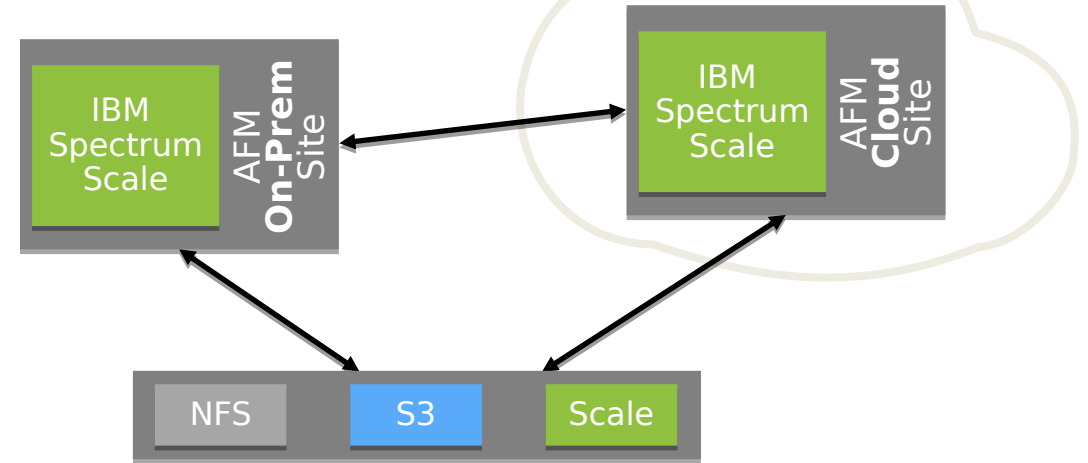
Data Caching Services (AFM) Use case details

Data Resilience



- Active-Passive DR over WAN or Cloud
- Designed for high latency and asynchronous DR
- Hot standby failover to DR site
- Automatic fallback data reconciliation
- Read-only access / analytics to all data at passive site

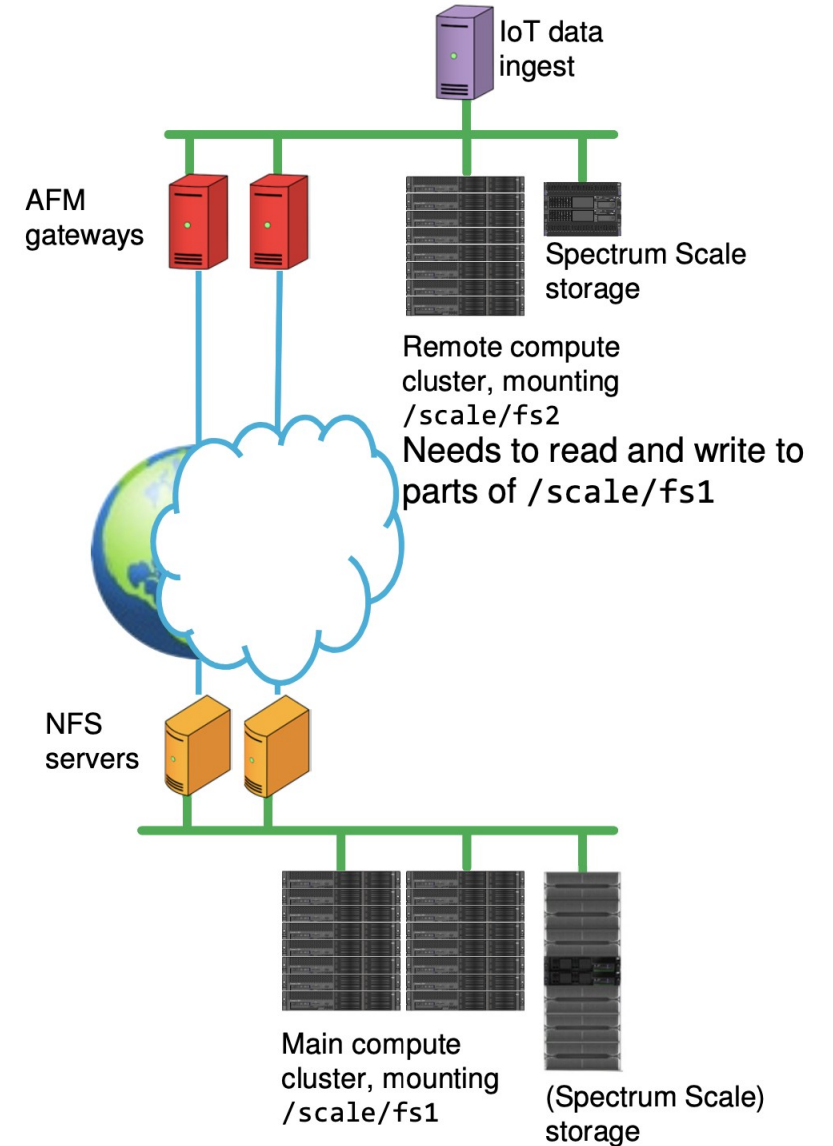
Hybrid cloud / Bursting



- Rapidly expand compute resources to cloud or data centers
- Common file system creates a single namespace across all locations
- Transparent access to data
- Cost effective way to increase compute on existing data
- Analytic results automatically pushed to home site

Data Caching Services: Active File management (AFM)

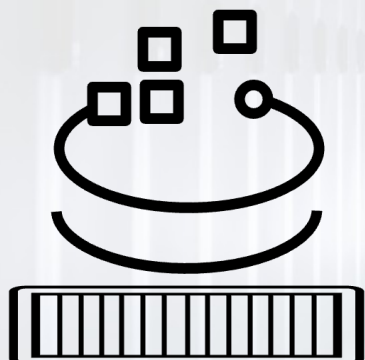
- Cache technology – Coherency, Currency across WAN
- Provides read & write caching across network outages
- Spans geographic distance and unreliable networks
 - Caches local 'copies' of data distributed to one or more Spectrum Scale clusters
 - Low latency 'local' read and write performance
 - As data is written or modified at one location, all other locations see that same data
- Configurable Cache
 - Cache size
 - Cache population strategies
 - Prefetch data upfront
 - Pull data on demand
- Asynchronous DR is a special case of AFM
 - Bidirectional awareness for Fail-over & Fail-back with data integrity
 - Recovery Point Objectives for volume & application consistency



Thank you for using



Storage Scale



Storage Scale
System