Spectrum Scale
Problem Determination

Please Note

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. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Deep Dive

Problem Determination

Problem Determination is the #1 concern I hear from all clients; large, small, old or new. The goal here is clear. I want clients to tell us at next year's User Groups - **"Wow, what you did helped me."**

Doris Conti | Director, Spectrum Scale

What We've Heard

"If we can't monitor something, we can't roll it out."

"I want meaningful alerts that don't cause alert fatigue. You can't tell the difference between a client leaving a cluster and a quorum node leaving a cluster."

"Our ops team is looking at dashboards all day. If something doesn't flash in red or come up on their monitoring console, they're not going to see it."

"One of the things that's really lacking in GPFS is constant monitoring."
"There are tens of thousands of components that could break at any given time."

"This is an art that you learn from experience."

"What is going on with my GPFS system?"

"What I really need is to be able to track down the rogue user who is bogging down the entire system."

Today

"There is really no clear way to understand what a healthy cluster looks like. If there is someone who knows, I'd love to talk to them."

Bob Oesterlin | **Nuance Communications**

Users rely on a wide variety of commands to monitor their Spectrum Scale cluster. This requires them to understand:

- Which components are important to monitor?
- Which commands should I use to monitor each component type?
- How do I interpret the results of all of the commands?

How to get the overall state of the system

Core GPFS

- mmgetstate → Daemon state / Quorum
- mmlsdisk → disk state
- mmdiag → Gpfs waiters
- /var/log/messages → FSSTRUCT Errors /var/adm/ras/mmfs.log.latest → detailed gpfs errors
- ....

Protocols

- Is Samba running? And CTDB?
- Is nfs-ganesha daemon responding?
- What about authentication daemons (SSSD, winbindd?)
- Are my Openstack services doing well?

And there are even more components to look at

- Network, AFM, Zimon, Backup, CCR, ...

Central State Command

mmhealth

A single CLI command that provides a health overview of all key components in the entire cluster.

```
$ mmhealth node show -v

Node name:      test_node
Node status:    degraded

Component       Status        Reasons
------------------------------------------
GPFSDaemon      healthy       -

CES             failed        smbd_down
      Auth          healthy       -
      OBJ_Auth      healthy       -
      NFS           healthy       -
      OBJ           healthy       -
      SMB           failed        smbd_down

Network         healthy       -

LocalDisk       healthy       -
      DiskA         healthy       -
      DiskB         healthy       -
      DiskC         healthy       -
      DiskD         healthy       -
      DiskE         healthy       -

FSMount         healthy       -
      FSI           healthy       -
      FSII          healthy       -
```

mmhealth

> Which components are important to monitor?

mmhealth will show all relevant components and group related components

> Which commands should I use to monitor each component type?

mmhealth will be the central point for getting the system state

> How do I interpret the results of all of these commands?

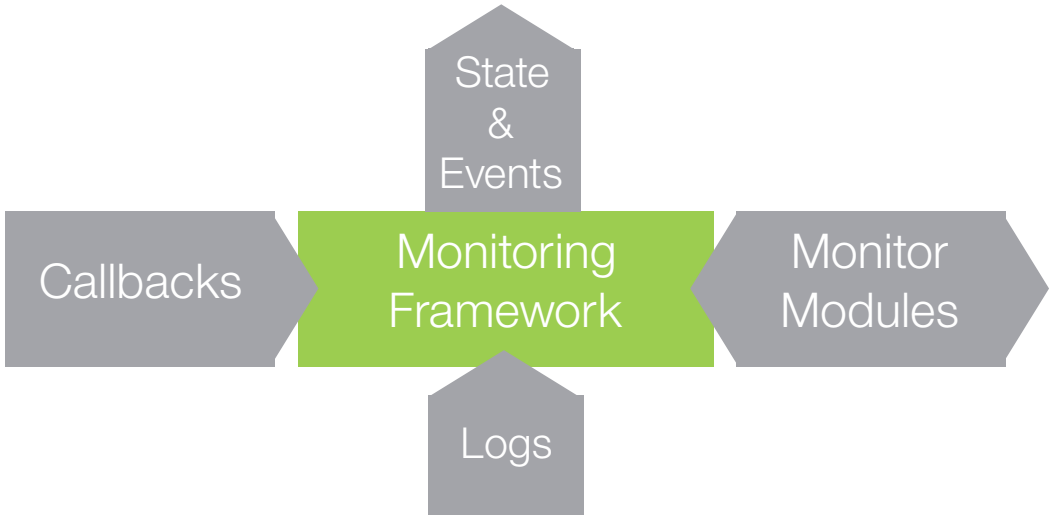
mmhealth shows a clear state for each component and the reason (event) for the state change.

```
Node name:      node003gpfs
Node status:    DEGRADED

Component       Status         Reasons
-------------------------------------------------
CES             FAILED         ctdb_recovery, ctdb_state_down
GPFS            HEALTHY        -
FILESYSTEM      FAILED         stale_mount
```

Monitoring Framework

A new monitoring component has been introduced with 4.1.1 on CES nodes.
It will be expanded to all gpfs cluster nodes



- Scalable to large number of nodes due to decentralized monitoring
- Well defined events and states
- Extensible
- Runs on Linux and AIX
- Part of GPFS ext package

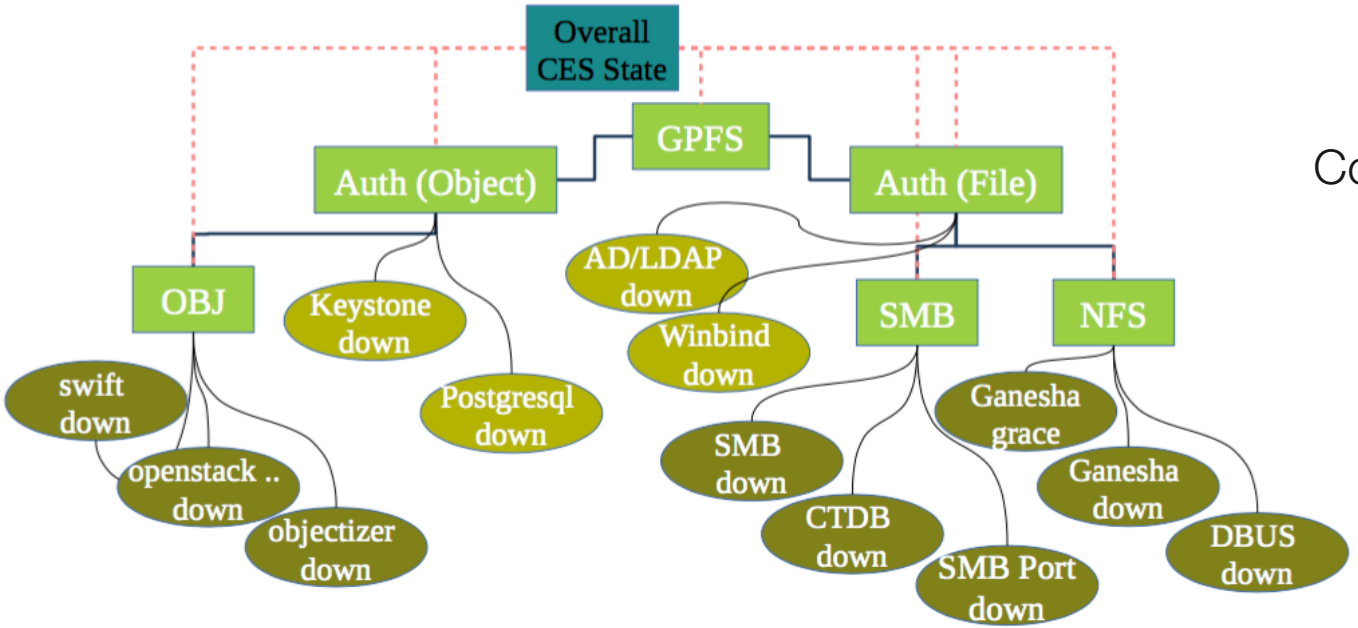
Events & States

Detect a problem → Raise well-defined event → Update component state

- Events contribute to the state of a component. If a component is unhealthy then looking at the events is the best way to get an idea of what is wrong
- Failure events always have a corresponding “good” event to clear the failure state automatically when the problem disappears

Events & States

Dependencies between component are taken into account → State DEPEND_FAILED



Component States

Events

mmhealth

Ability to drill down into component details

```
$ mmhealth node show gpfs -v

Node name:                      node003gpfs

Component                       Status        Reasons
-------------------------------------------------------
GPFSDaemon                      healthy       -

Event                           Parameter     Severity          Description
-------------------------------------------------------------------------------
gpfs_up                         GPFS          INFO              GPFS process now running
gpfsport_up                     GPFS          INFO              GPFS port 1191 is active
longwaiters_found_down          GPFS          INFO              No GPFS long-waiters
quorum_up                       GPFS          INFO              Quorum detected
```

Option -v/--verbose shows “good” events to see what is functioning well

Option --unhealthy provides the ability to filter output for non-healthy components

mmhealth

See the event history, useful tool for identifying what caused an issue that has been recovered since then.

```
$ mmhealth node eventlog

Timestamp                 Event Name      Severity     Details
-------------------------------------------------------------------------
2016-03-08 03:26:30 EST   ctdb_recovery   WARNING      CTDB Recovery detected
```

All events are also pushed to the syslog by default.

Event Notification

Spectrum Scale already supports sending notifications over SNMP or email

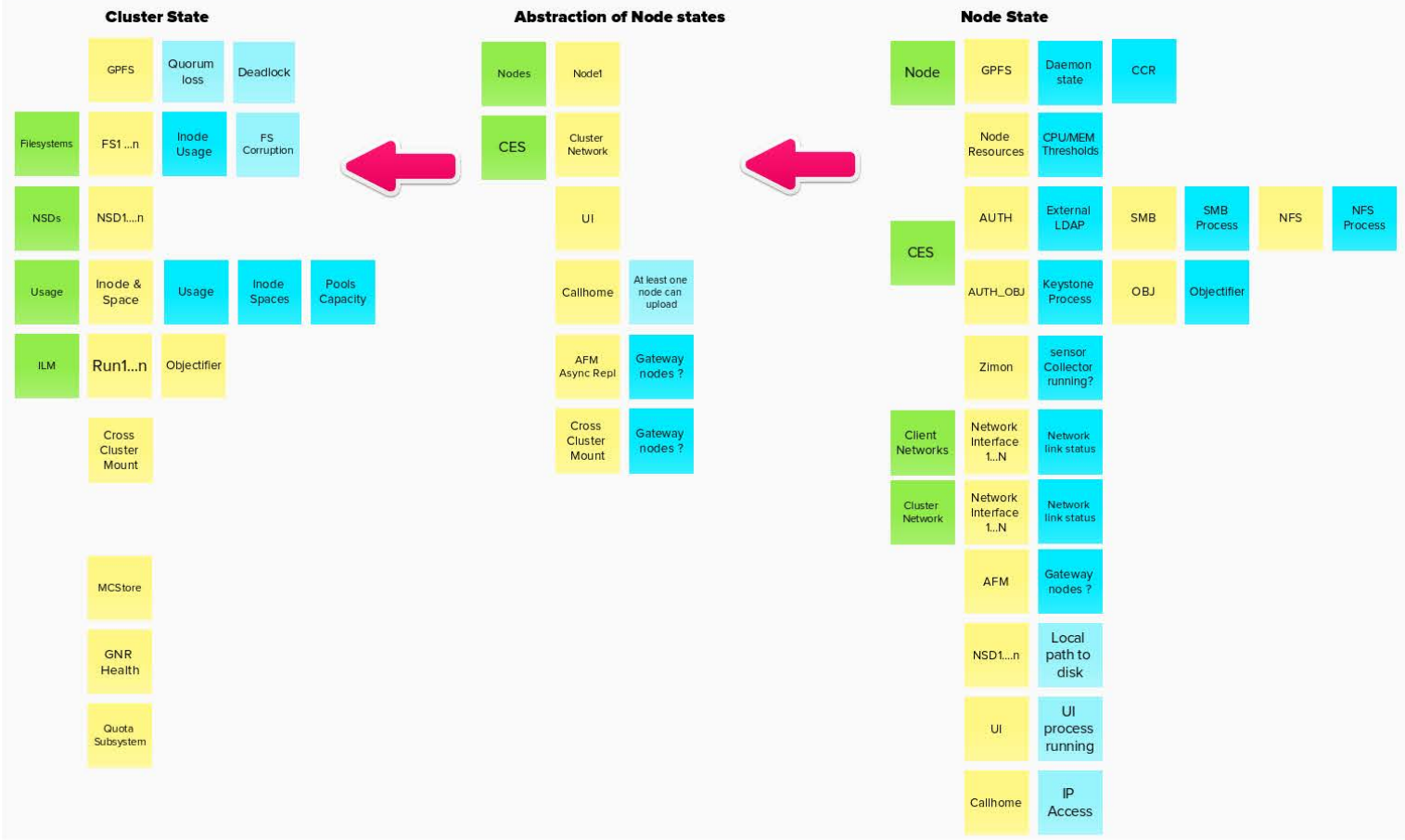
- GUI can send SNMP Traps for each event
- GUI can send e-mails for each event
- GPFS SNMP Subagent can send traps for particular GPFS issues
    - SNMP queries are supported for a small subset

In future releases we plan to consolidate the event notification inside the health monitoring.

- Single point of configuration for SNMP,Email
- No dependency on GUI
- Add additional notification methods and plugins interfaces (e.g. nagios)

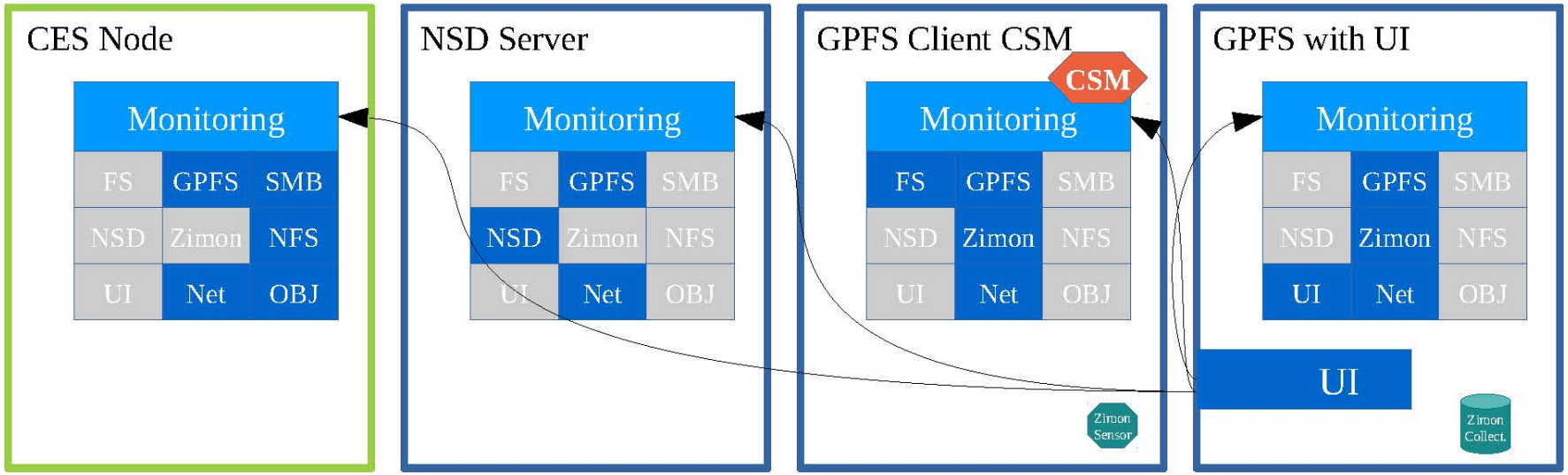
Monitoring

Monitoring - Development Sketch



Monitoring - Node Roles

The role of a node will determine what components need to be monitored.



Monitoring - GPFS

Detecting GPFS problems by listening to system callbacks and active monitoring of the GPFS daemon.

The GPFS Monitor will run on any cluster node and detect issues like:

- Daemon State (Active/Down/Arbitrating)
- Daemon Network Port
- Quorum Loss
- Node Expel
- Deadlocks/Critical Long Waiters
- CCR State
- Configuration inconsistencies

Monitoring - Filesystem

Detecting issues with any filesystem by listening to system callbacks and active monitoring of the filesystem mount state.

The Filesystem monitor will run on any GPFS client. It will depend on the GPFS daemon state and detect the following:

Node Level

- Unexpected unmount (for example, FS Panic)
- Filesystem corruption (FS_Struct Errors)
- Inconsistent mount state

Cluster Level

- Filesystem III-replication
- Filesystem descriptor quorum

Monitoring - NSDs

Detecting issues with any NSD in the system by listening to system callbacks and active monitoring of the disk state.

The NSD monitor will run on NSD servers only. It depends on the GPFS daemon state. It will monitor things like:

- Availability of the disk
- Multipathing
- Physical disk state

A broken NSD state will also change the corresponding filesystem state.

Monitoring - Network

The cluster network as well as the client network (CES) will be monitored. It will monitor the network interfaces which are used by Spectrum Scale.

The cluster network monitor will run on any gpfs node while the client network monitoring runs on CES nodes only.

- Per NIC State
- Bonding state
- Infiniband state
- IP Connectivity
- DNS
- Thresholds on TCP error counts

Monitoring - Protocols

Protocol monitoring has been introduced with 4.1.1 already. It monitors all enabled protocol components.

The protocol monitors will run on CES nodes only. It will monitor several components like:

SMB

- SMB Daemon & Port
- CTDB Daemon Status & Recovery

NFS

- nfs-ganesha daemon
- Portmapper, statd (v3), DBUS

Object

- Openstack processes , PostgreSQL
- Ringfile checksum

Monitoring - Authentication

Authentication monitoring is part of the protocol monitoring and has been introduced with 4.1.1 already. With 4.2.0 monitoring of external authentication servers has been added.

The authentication monitor will run on CES nodes and monitor:

Active Directory Authentication
- Winbindd process / join state
- Auth Server connectivity

LDAP Authentication
- SSSD process
- LDAP Server connectivity

NIS
- ypbind service
- NIS Server connectivity

Keystone service (Object authentication)
- Connectivity to external keystone

Monitoring - Zimon

The performance monitoring daemon will be monitored actively ....

The Zimon monitor will run on sensor nodes and collector nodes.

It will monitor things like:

- Collector daemon up and running
- Zimon sensors operational

Monitoring - AFM

Active File Management (AFM) is used for disaster recovery (DR) and WAN caching.

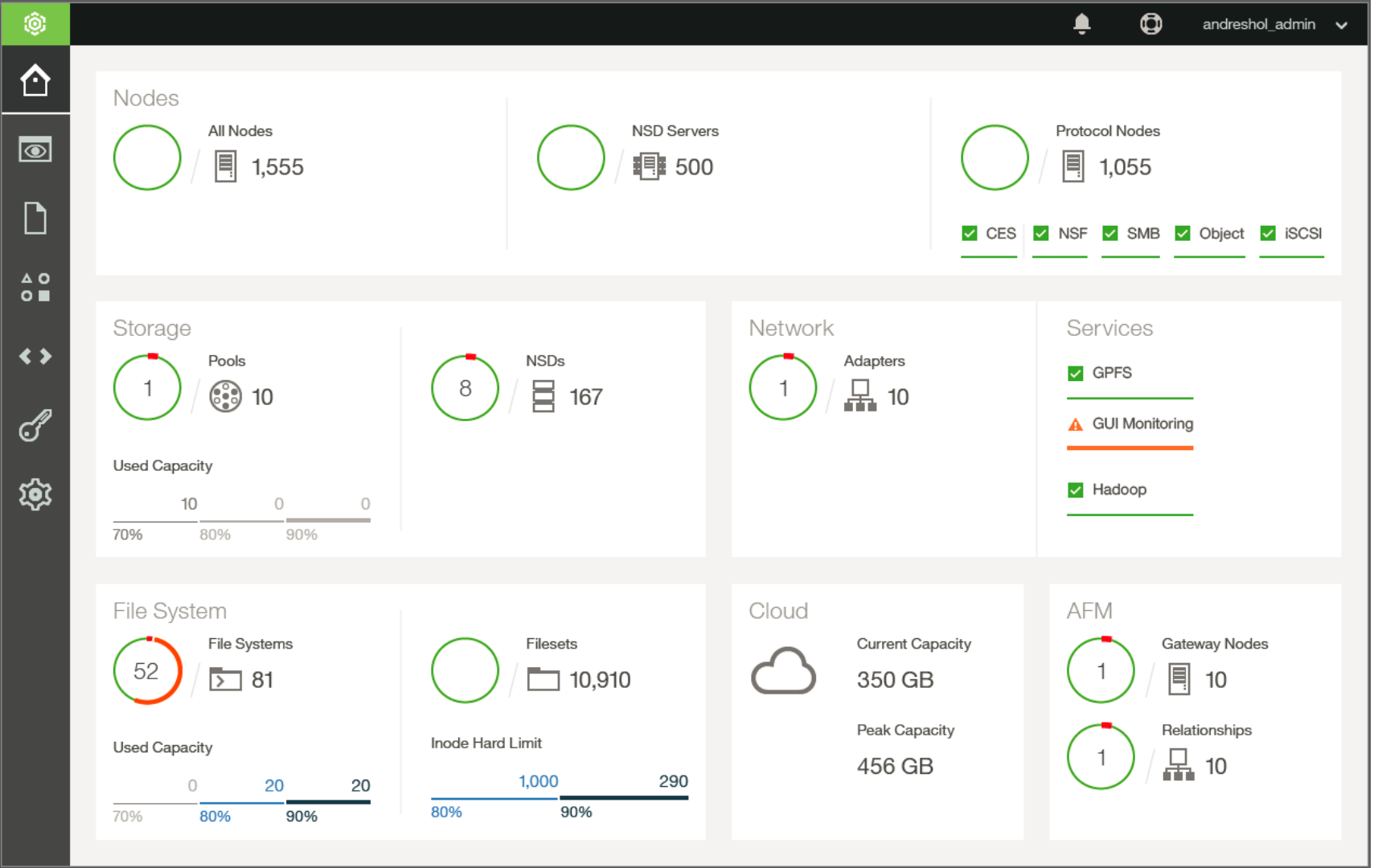
The AFM monitor will run on AFM gateway nodes and monitor:

- AFM Gateway state
- Site connected/disconnected
- Queue overflow/drop
- RPO missed

Monitoring … more

The monitoring will get events from more components. It will be extended over time to cover additional components and failure conditions.

- TSM Backup
- DMAPI
- Hadoop Namenode/datanode
- Cloud tiering
- Callhome
- NTP
- GUI
- Native Raid
- Resource Usage
- .....

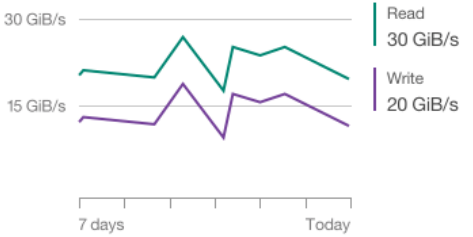

andreshol_admin
Nodes
All Nodes
1,555
NSD Servers
500
Protocol Nodes
1,055
CES
NSF
SMB
Object
iSCSI
Storage
1
Pools
10
8
NSDs
167
Used Capacity
10
0
0
70%
80%
90%
Network
1
Adapters
10
Services
GPFS
GUI Monitoring
Hadoop
File System
52
File Systems
81
Filesets
10,910
Used Capacity
0
20
20
70%
80%
90%
Inode Hard Limit
1,000
290
80%
90%
Cloud
Current Capacity
350 GB
Peak Capacity
456 GB
AFM
1
Gateway Nodes
10
1
Relationships
10

admin

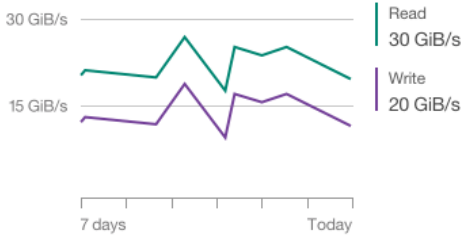
Nodes

Last 7 days

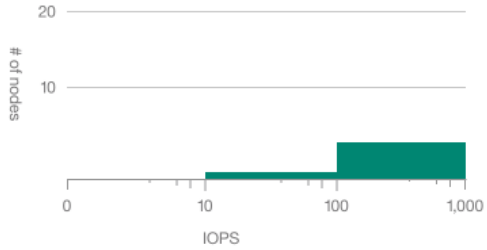
Overall Client Workload | Data Rate



Overall NSD Server | Data Rate



All Nodes by | Data Rate

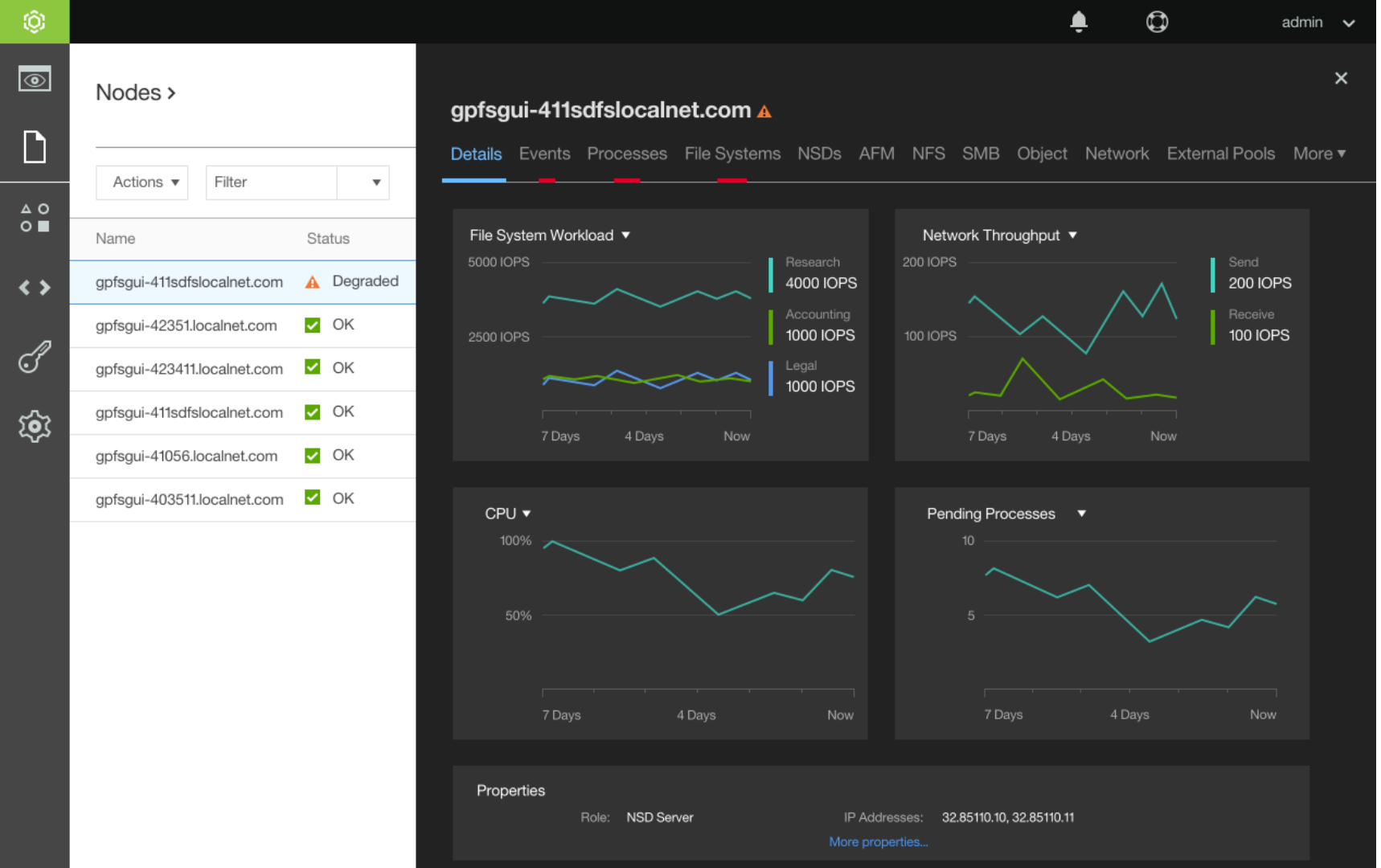


All Nodes | NSD Servers | Protocol Nodes

View Details

Filter

Name	Status	CPU Utilization	Load	Total Throughput	Response Time	Protocol
gpfsgui-411sdfslocalnet.com	Degraded	10%	55,555	100 MiB/s	30 ms	
gpfsgui-42351.localnet.com	OK	30%	500	500 MiB/s	7 ms	✓
gpfsgui-423411.localnet.com	OK	30%	500	500 MiB/s	7 ms	
gpfsgui-411sdfslocalnet.com	OK	30%	500	500 MiB/s	7 ms	
gpfsgui-41056.localnet.com	OK	30%	500	500 MiB/s	7 ms	
gpfsgui-403511.localnet.com	OK	30%	500	500 MiB/s	7 ms	


admin
Nodes >
Actions
Filter
Name
Status
gpfsgui-411sdfslocalnet.com
Degraded
gpfsgui-42351.localnet.com
OK
gpfsgui-423411.localnet.com
OK
gpfsgui-411sdfslocalnet.com
OK
gpfsgui-41056.localnet.com
OK
gpfsgui-403511.localnet.com
OK
gpfsgui-411sdfslocalnet.com
Details
Events
Processes
File Systems
NSDs
AFM
NFS
SMB
Object
Network
External Pools
More
File System Workload
5000 IOPS
2500 IOPS
7 Days
4 Days
Now
Research
4000 IOPS
Accounting
1000 IOPS
Legal
1000 IOPS
Network Throughput
200 IOPS
100 IOPS
7 Days
4 Days
Now
Send
200 IOPS
Receive
100 IOPS
CPU
100%
50%
7 Days
4 Days
Now
Pending Processes
10
5
7 Days
4 Days
Now
Properties
Role:
NSD Server
IP Addresses:
32.85110.10, 32.85110.11
More properties...

admin

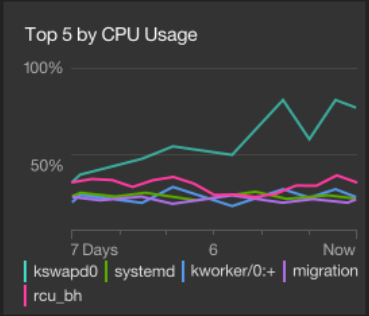
Nodes >

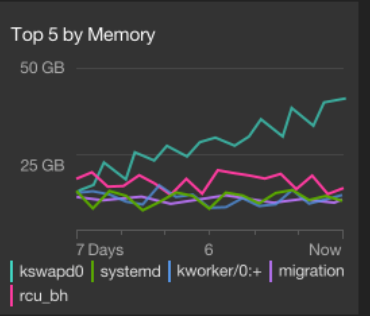
Actions ▾ | Filter ▾

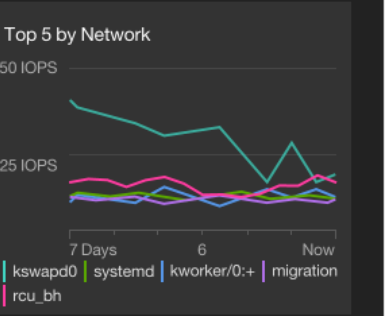
Name	Status
gpfsgui-411sdfslocalnet.com	⚠ Degraded
gpfsgui-42351.localnet.com	✓ OK
gpfsgui-423411.localnet.com	✓ OK
gpfsgui-411sdfslocalnet.com	✓ OK
gpfsgui-41056.localnet.com	✓ OK
gpfsgui-403511.localnet.com	✓ OK

gpfsgui-411sdfslocalnet.com ⚠

Details | Events | Processes | File Systems | NSDs | AFM | NFS | SMB | Object | Network | External Pools | More ▾







Process	Status	Type	CPU	Last 24 Hours	Network	User
Keystone	⚠ Not Responding	System	0%		0 IOPS	Nathan Cheshire
Object Storage	✓ Running	System	23%		20 IOPS	Pace Horton
Compression	✓ Running	Compression	21%		20 IOPS	Chester Womack
kswapd0	✓ Running	System	19%		20 IOPS	Abner Fairchild
systemd	✓ Running	System	27%		20 IOPS	Peyton Hope
kworker/0:+	✓ Running	Replication	15%		20 IOPS	Joann Haines
migration	✓ Running	System	18%		20 IOPS	Laurelle Newton
rcu_bh	✓ Running	Compression	20%		20 IOPS	Heath Josephson

admin

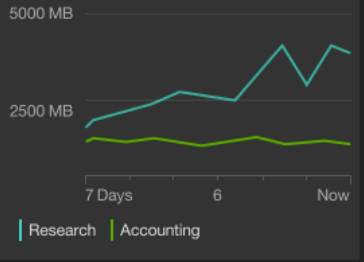
Nodes >

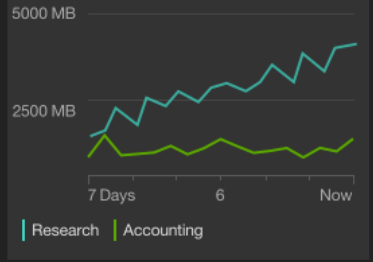
Actions ▾ | Filter ▾

Name	Status
gpfsgui-411sdfslocalnet.com	Degraded
gpfsgui-42351.localnet.com	OK
gpfsgui-423411.localnet.com	OK
gpfsgui-411sdfslocalnet.com	OK
gpfsgui-41056.localnet.com	OK
gpfsgui-403511.localnet.com	OK

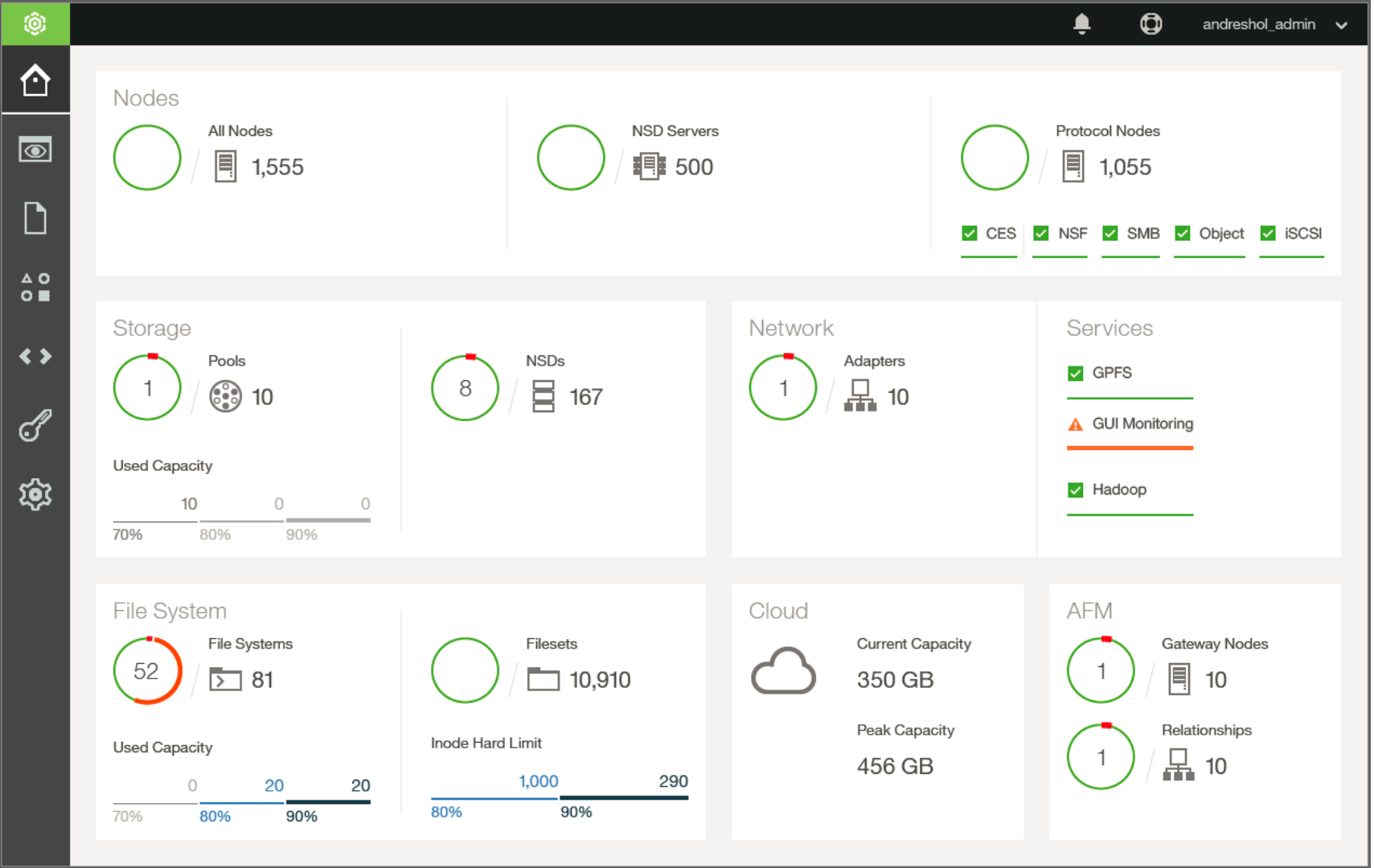
gpfsgui-411sdfslocalnet.com

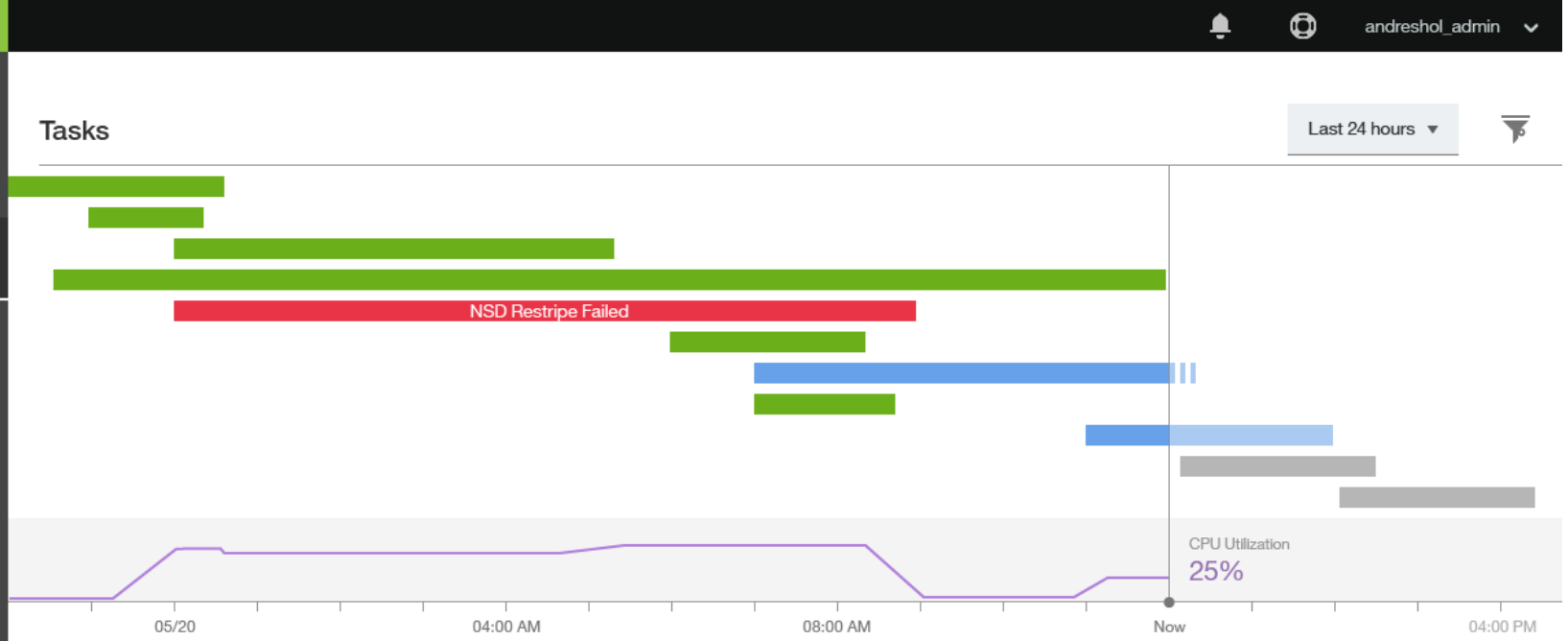
Details Events Processes File Systems NSDs AFM NFS SMB Object Network External Pools More ▾





Name	Status	Read Rate	Write Rate	IO Rate	Response Time	Capacity
Accounting_123	Degraded	2.5 GB/s (30%)	1.0 GB/s (35%)	3000 IOPS	100 ms	75%
Accounting	Mounted	2.5 GB/s (45%)	0.7 GB/s (100%)	3000 IOPS	100 ms	81%
Legal	Not Mounted					52%


andreshol_admin
Nodes
All Nodes
1,555
NSD Servers
500
Protocol Nodes
1,055
CES
NSF
SMB
Object
iSCSI
Storage
1
Pools
10
8
NSDs
167
Used Capacity
10
0
0
70%
80%
90%
Network
1
Adapters
10
Services
GPFS
GUI Monitoring
Hadoop
File System
52
File Systems
81
Filesets
10,910
Used Capacity
0
20
20
70%
80%
90%
Inode Hard Limit
1,000
290
80%
90%
Cloud
Current Capacity
350 GB
Peak Capacity
456 GB
AFM
1
Gateway Nodes
10
1
Relationships
10


andreshol_admin
Tasks
Last 24 hours
NSD Restripe Failed
CPU Utilization
25%
05/20
04:00 AM
08:00 AM
Now
04:00 PM
Alerts
01:00 AM NSD Restripe Failed
Currently Running Tasks
2

Problem Determination

Thresholds

Administrator want the ability to set thresholds so lower level operations teams can assess if a value is a problem or not.

The performance monitoring framework will support thresholds to be configured for any metric.

- Predefined thresholds will be used to create monitoring events e.g.
  - Filesystem and disk usage
  - Memory usage
- A user can create thresholds for any metric and be notified if the threshold is hit

Declustered Arrays

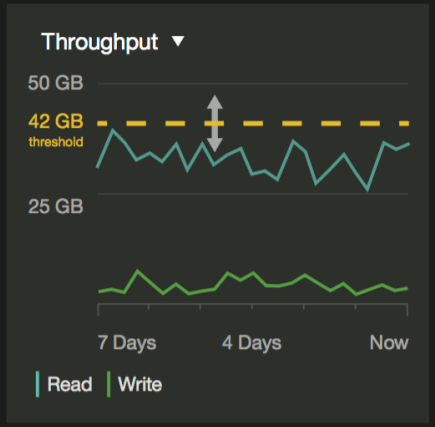
Actions ▾ | Filter | ▾

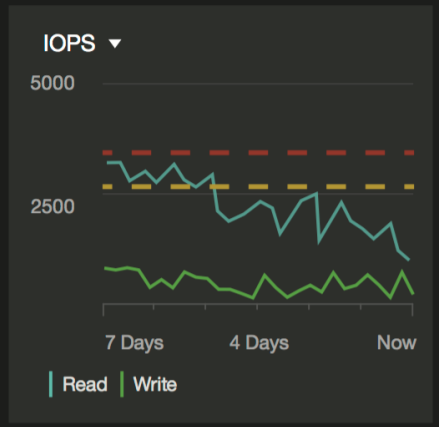
Name	Status
LOG	OK
DA1	1 pDisk failed
DA2	OK

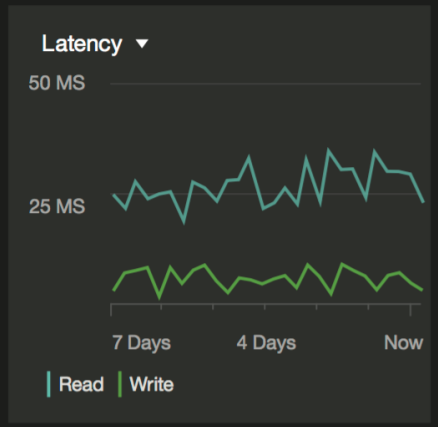
DA1

50 TiB used — 100 TiB total

Details | Events | Virtual Disks | Physical Disks







Properties

Recovery Group: BB1RGL
Scrub Duration: 14
Replace Threshold: 2
Spare Count: 2
Free Space: 1.7 TiB

More properties...

Long Waiters

Looking at GPFS long waiters can help to understand the system load and find certain bottlenecks.

In addition to the command line (mmdiag), long waiters will be available through the performance monitoring interface (mmpmon/Zimon).

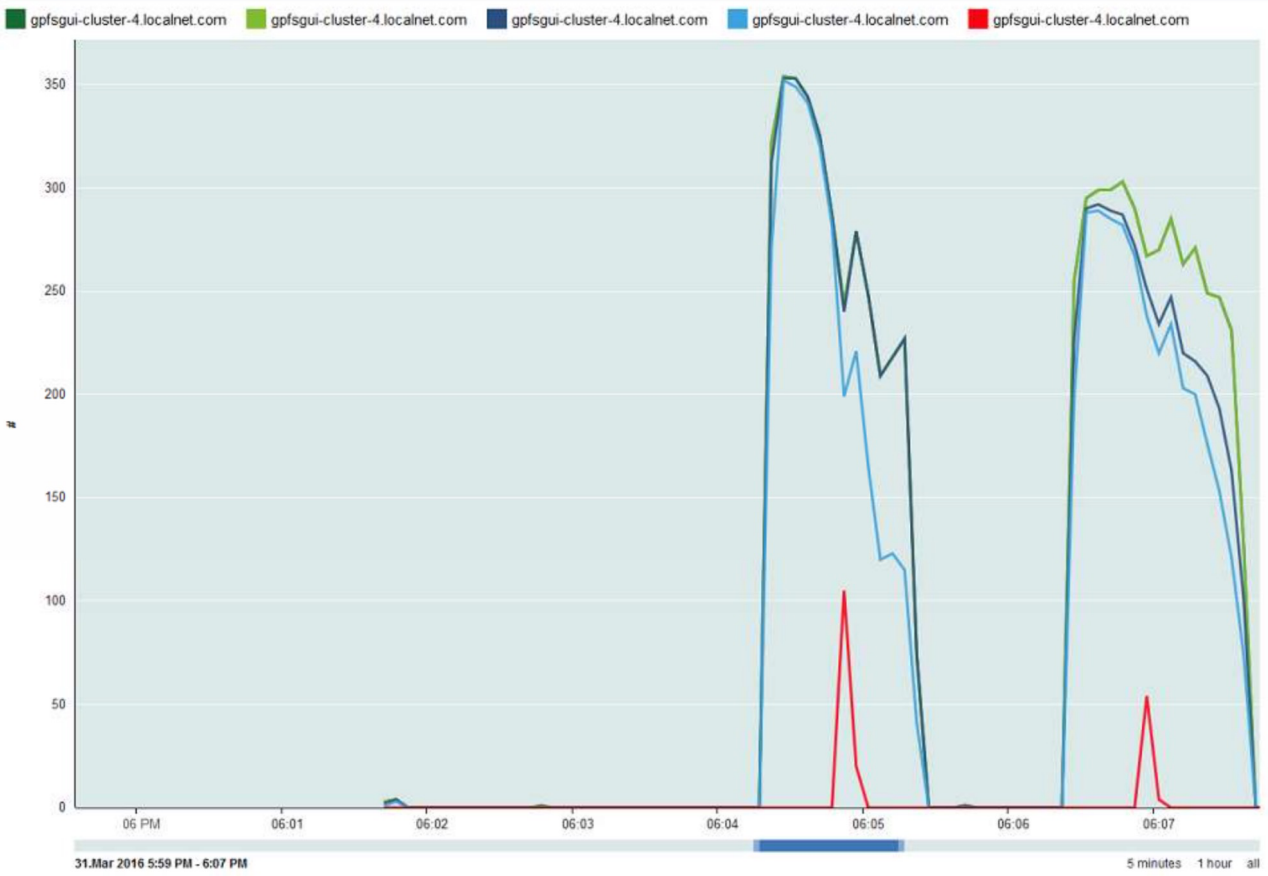
- Categorize waiters
  - For example, Disk IO vs Network
- UI can visualize different waiter category counts or long waiter counts in a timeline
- Show long waiter counts side by side with other metrics (for example, throughput)

admin

Performance

Search

Cluster / IBM Spectrum Scale™ Client / GPFS Waiters / All Waiters



Resources

Resource type: Waiters

Aggregation level: Cluster

Filter: *no filter*

5 minutes 1 hour 1 day 1 week 1 month 1 year all

Search

Name	All Waiters
0.1s	2515 #
0.2s	2514 #
0.5s	2499 #
1.0s	2339 #
30.0s	125 #
60.0s	0 #
all	2731 #

Metrics

The performance metrics defined in the performance tool helps to collect the performance data based on various aspects. Learn more

IBM Spectrum Scale™ Client

GPFS Waiters :

- [x] All Waiters
- [ ] Waiters (Local Disk IO)
- [ ] Waiters (Network IO)
- [ ] Waiters (ThCond)
- [ ] Waiters (ThMutex)
- [ ] Waiters (Delay)
- [ ] Waiters (Syscall)

Apply Close

Questions?