IBM Spectrum Storage

Spectrum Scale

Problem Determination

Mathias Dietz

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.

Problem Determination

Problem Determination is a major focus area in Spectrum Scale development in 2016.

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

Problem Determination Scope

Monitoring

- Automatically detect problems in the Spectrum Scale system
- Find the “root cause” of a problem
- Hide system complexity → easy to understand

Problem Isolation

- Collect all required informations to analyse problems
- Provide tools to help customers/support to isolate problems
- Provide metrics to better understand system load

Problem Recovery

- Automatically repair problems where possible
- Guide customers through troubleshooting process
- Ensure System availability through failover/recovery

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.”

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?

Problem Determination is sometimes like .....

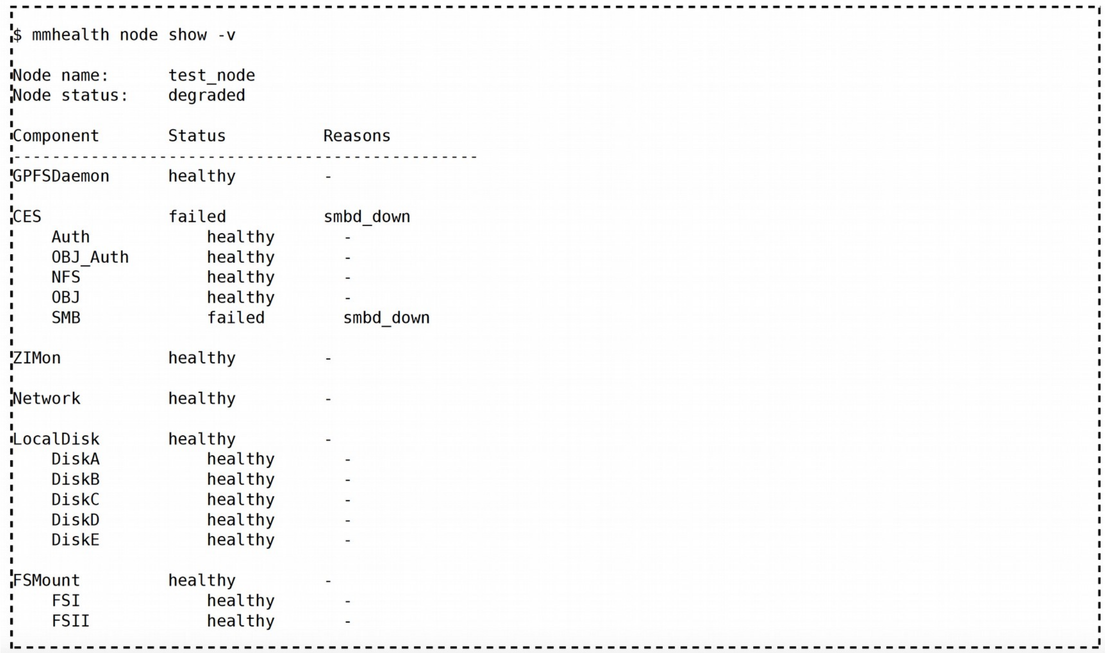
How to get the overall state of the system ?

- **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

ZIMon           healthy         -

Network         healthy         -

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

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

Central State command – mmhealth details

>Which components are important to monitor?

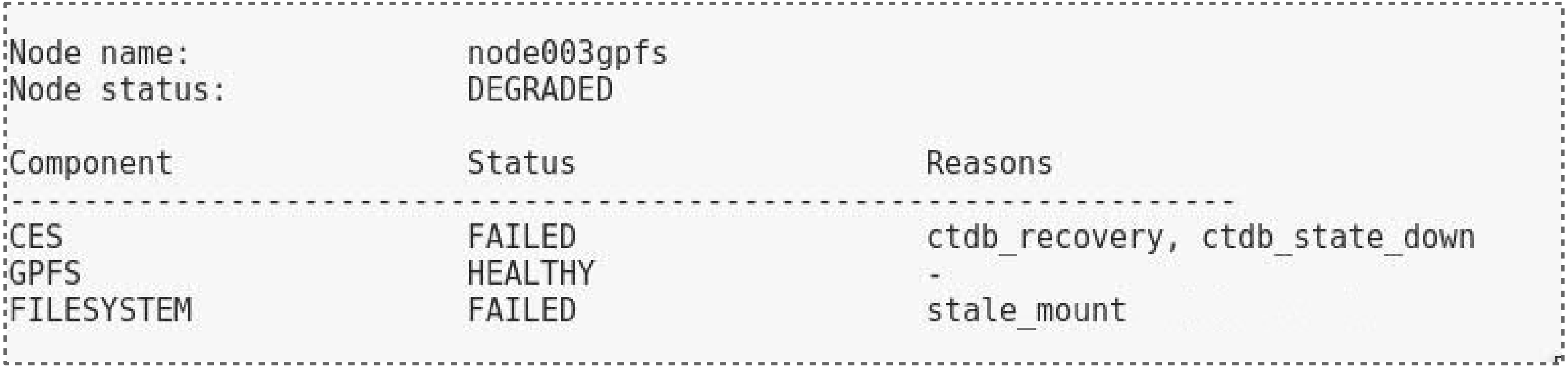
mmhealth will show all relevant components and group related components

>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 the commands?

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

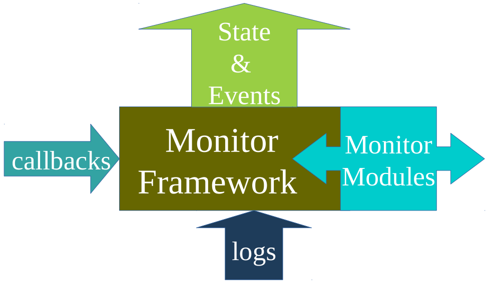


```
Node name:                node003gpfs
Node status:              DEGRADED

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

Monitoring Framework

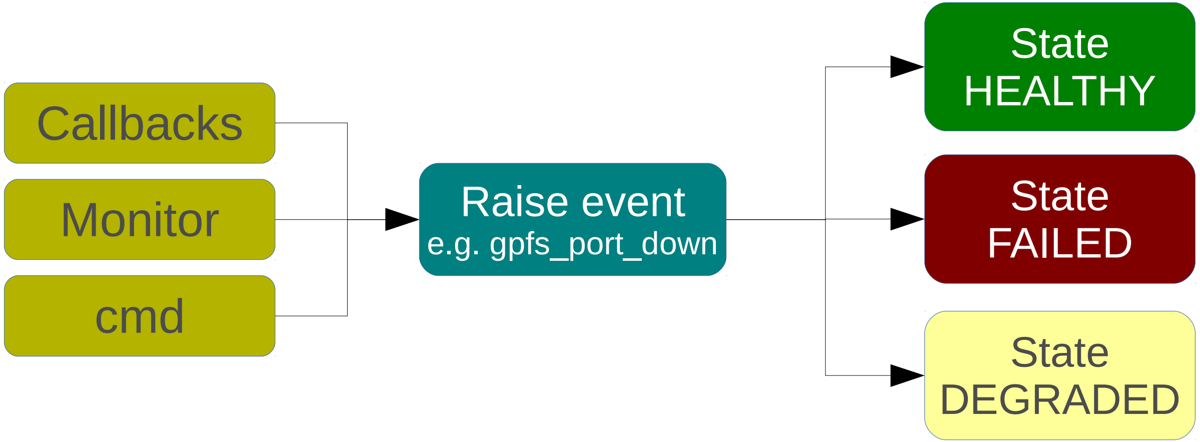
A new monitoring component has been introduced with 4.1.1 on CES nodes. It will be expanded to all gpfs cluster nodes

- 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



Central State command – Events and States

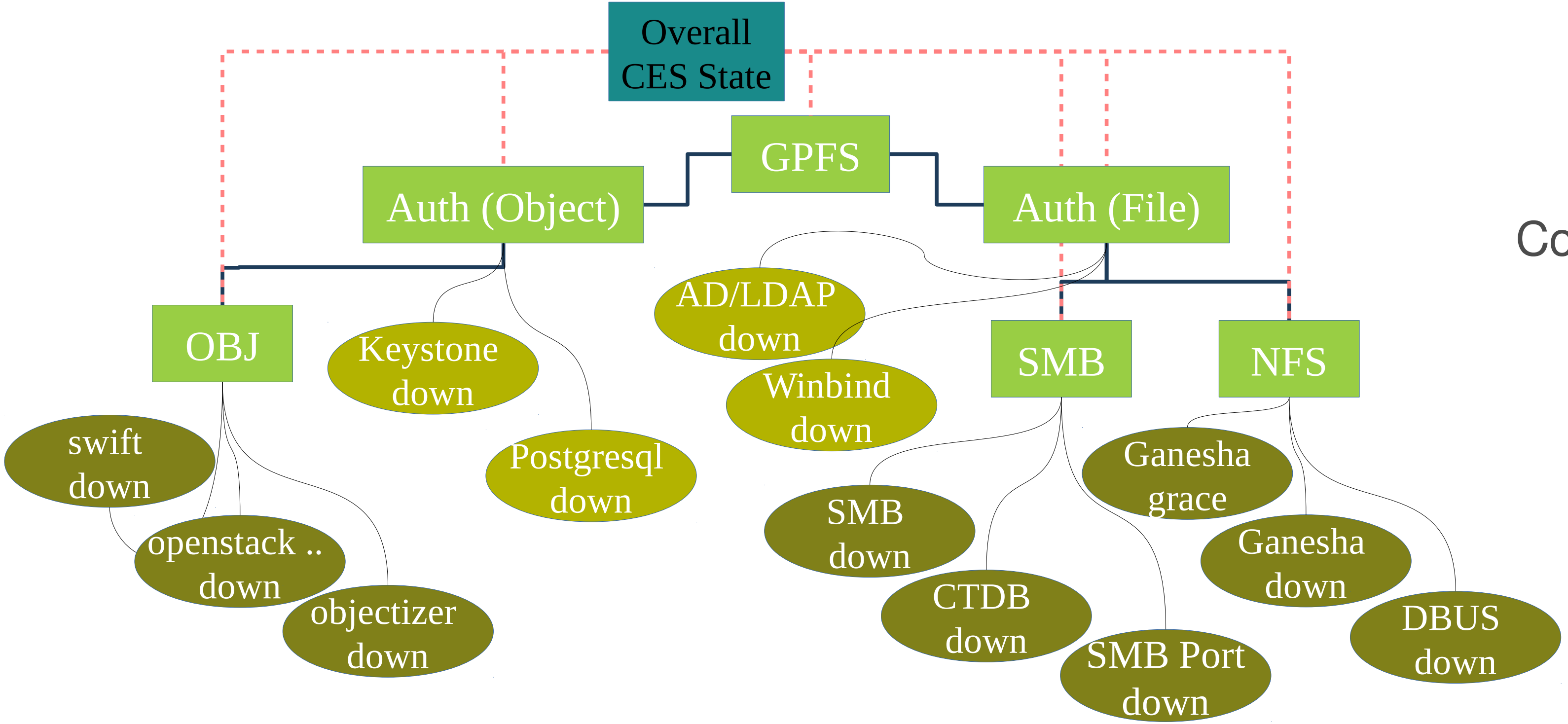
Detect a problem → raise well defined event → update component state



- 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

Central State command – Events and States

Dependencies between component are taken into account → State DEPEND_FAILED



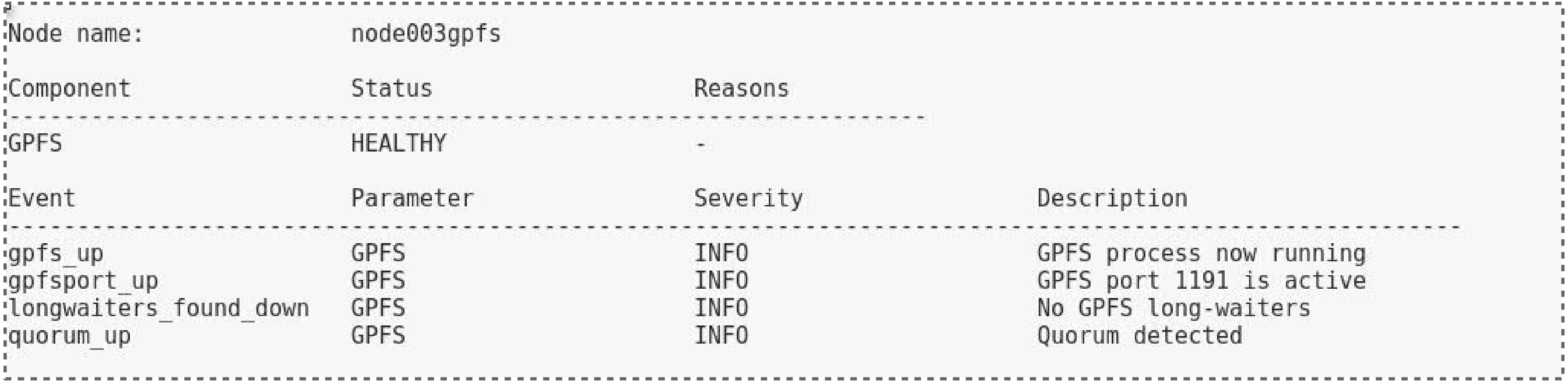
Component States

Events

Central State command - mmhealth

Ability to drill down into component details

#> mmhealth node show gpfs -v



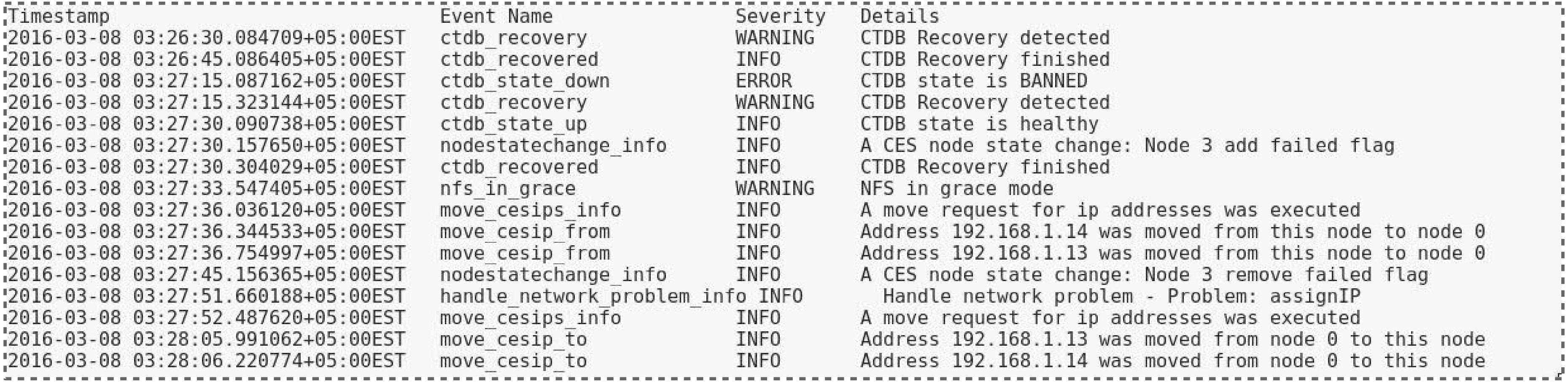
```
Node name:              node003gpfs

Component               Status                  Reasons
--------------------------------------------------------------------
GPFS                    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

Central State command - 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 03:26:30.084709+05:00EST   ctdb_recovery              WARNING    CTDB Recovery detected
2016-03-08 03:26:45.086405+05:00EST   ctdb_recovered             INFO       CTDB Recovery finished
2016-03-08 03:27:15.087162+05:00EST   ctdb_state_down            ERROR      CTDB state is BANNED
2016-03-08 03:27:15.323144+05:00EST   ctdb_recovery              WARNING    CTDB Recovery detected
2016-03-08 03:27:30.090738+05:00EST   ctdb_state_up              INFO       CTDB state is healthy
2016-03-08 03:27:30.157650+05:00EST   nodestatechange_info       INFO       A CES node state change: Node 3 add failed flag
2016-03-08 03:27:30.304029+05:00EST   ctdb_recovered             INFO       CTDB Recovery finished
2016-03-08 03:27:33.547405+05:00EST   nfs_in_grace               WARNING    NFS in grace mode
2016-03-08 03:27:36.036120+05:00EST   move_cesips_info           INFO       A move request for ip addresses was executed
2016-03-08 03:27:36.344533+05:00EST   move_cesip_from            INFO       Address 192.168.1.14 was moved from this node to node 0
2016-03-08 03:27:36.754997+05:00EST   move_cesip_from            INFO       Address 192.168.1.13 was moved from this node to node 0
2016-03-08 03:27:45.156365+05:00EST   nodestatechange_info       INFO       A CES node state change: Node 3 remove failed flag
2016-03-08 03:27:51.660188+05:00EST   handle_network_problem_info INFO        Handle network problem - Problem: assignIP
2016-03-08 03:27:52.487620+05:00EST   move_cesips_info           INFO       A move request for ip addresses was executed
2016-03-08 03:28:05.991062+05:00EST   move_cesip_to              INFO       Address 192.168.1.13 was moved from node 0 to this node
2016-03-08 03:28:06.220774+05:00EST   move_cesip_to              INFO       Address 192.168.1.14 was moved from node 0 to this node
```

All events are also pushed to the syslog by default.

Event Notificaton

Spectrum Scale already supports sending notifications over SNMP or email

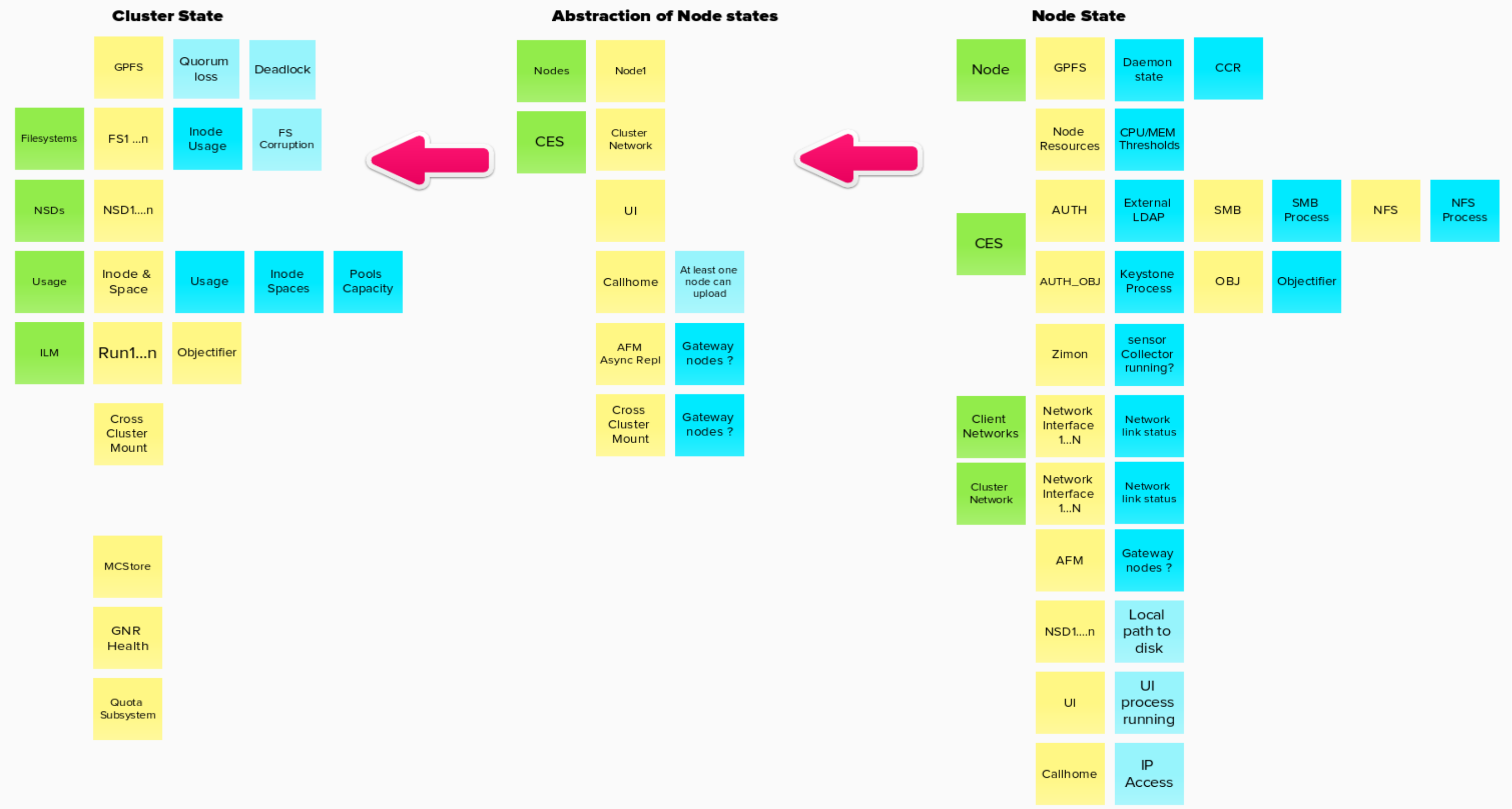
- GUI can send SNMP Traps for each event
- GUI can send emails for each event
- GPFS SNMP Subagent can send traps for particular GPFS issues
  - SNMP queries are supported for a small subset

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



Cluster State

	GPFS	Quorum loss	Deadlock	
Filesystems	FS1 ...n	Inode Usage	FS Corruption	
NSDs	NSD1....n			
Usage	Inode & Space	Usage	Inode Spaces	Pools Capacity
ILM	Run1...n	Objectifier		
	Cross Cluster Mount			
	MCStore			
	GNR Health			
	Quota Subsystem			

Abstraction of Node states

Nodes	Node1	
CES	Cluster Network	
	UI	
	Callhome	At least one node can upload
	AFM Async Repl	Gateway nodes ?
	Cross Cluster Mount	Gateway nodes ?

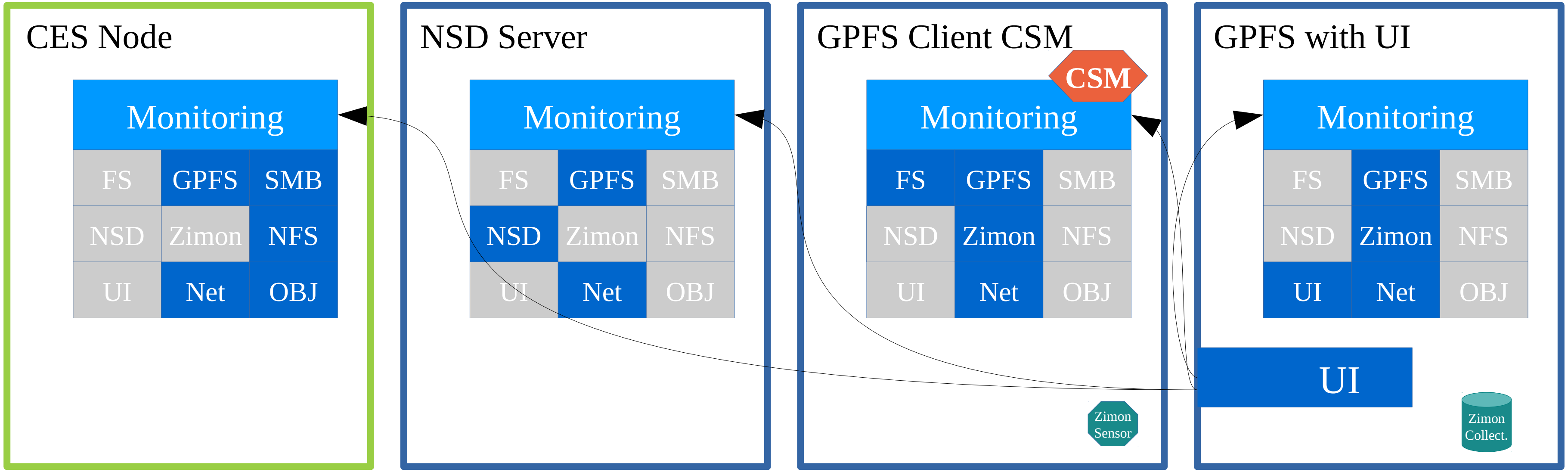
Node State

Node	GPFS	Daemon state	CCR			
	Node Resources	CPU/MEM Thresholds				
CES	AUTH	External LDAP	SMB	SMB Process	NFS	NFS Process
	AUTH_OBJ	Keystone Process	OBJ	Objectifier		
	Zimon	sensor Collector running?				
Client Networks	Network Interface 1...N	Network link status				
Cluster Network	Network Interface 1...N	Network link status				
	AFM	Gateway nodes ?				
	NSD1....n	Local path to disk				
	UI	UI process running				
	Callhome	IP Access				

IBM Spectrum Storage

Health Monitoring – Node Roles

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



GPFS Monitor

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
- Memory allocation
- Configuration inconsistencies

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 un-mounts (e.g. FS Panic)
- Filesystem corruption (FS_Struct Errors)
- Inconsistent mount state

Cluster level

- Filesystem Ill-replication
- Filesystem descriptor quorum

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

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

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

Authentication

Authenitcation 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

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

AFM

Active File Management (AFM) is used for desaster 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

and 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
.....

Problem Isolation

Active Directory Tool

Customers have trouble integrating Spectrum Scale in their Authentication environment because of unmet requirements.

Today problem determination requires expert skills

Users can verify that the authentication environment of the customer fulfills the requirements of Spectrum Scale.

Connectivity to Authentication servers
Find connectivity issues to domain controllers, e.g. caused by wrong firewall
Complex trust relationships
Verify multi-domain trust relationships and detect unsupported configs
ID Mapping inconsistencies
Verify required UID/GID fields are filled and fit into configured id ranges to isolate the root cause of access failures.

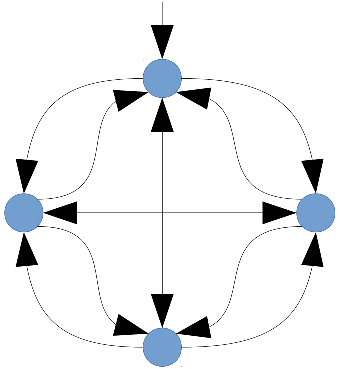
Network Verification

"When we have issues and we're pretty sure it is the network, we still have to spend however many hours to write a test case that doesn't involve GPFS to prove that it is exclusively the network. A network verification tool would be a big help."

Verify node to node connectivity to detect to common network issues

General connectivity issues
e.g. broken IP/Routing/Switch config
Firewall configuration issues
*e.g. some nodes could *not* reach others on 1191 *or* on any of the ephemeral ports.*
Network Performance issues
Low throughput / High Latency

Network Verification



mmnetverify command will verify the network connectivity for a given list of nodes.

- Connect to other nodes through ssh
- Spawn netverify daemon on each node
- Verify port connectivity from any to any node (GPFS port + ephemeral port range)
- Verify IB connectivity from any to any node
- Generate reference workload and evaluate network performance (latency/throughput)

Tracing

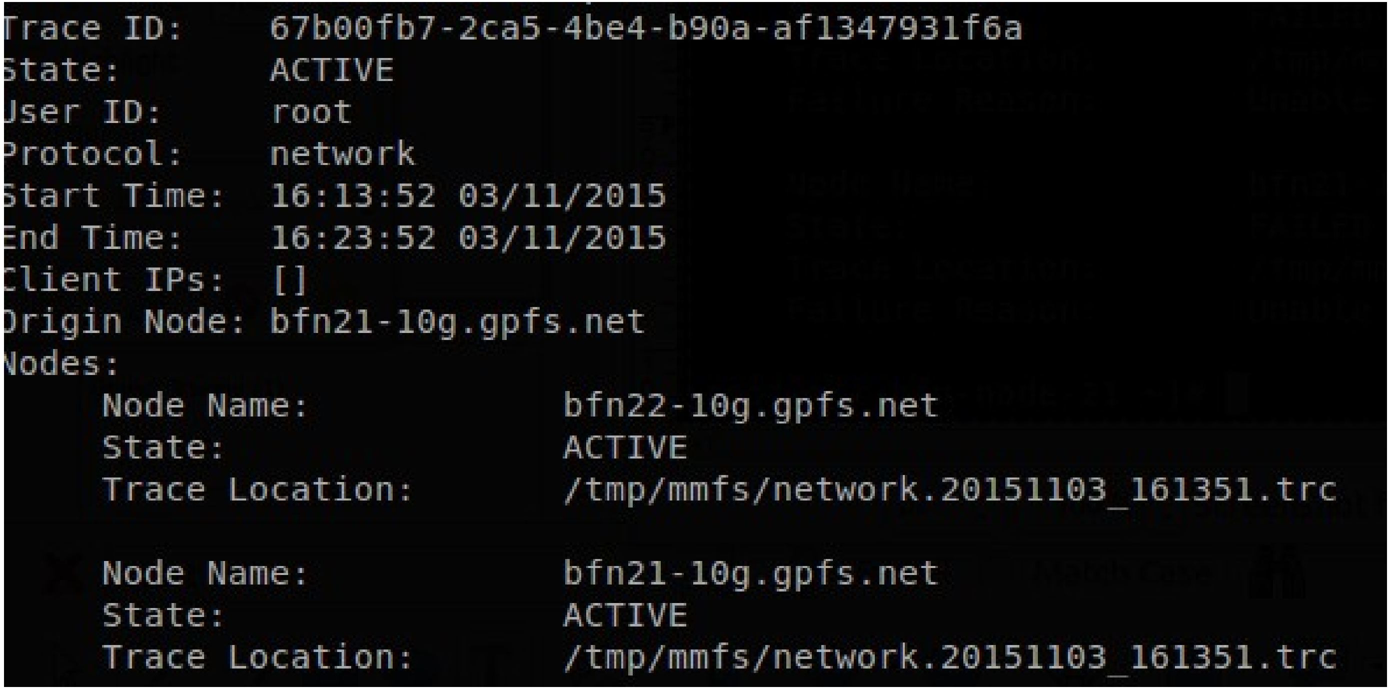
In addition to the GPFS tracing facitily, protocol tracing has been introduced in 4.1.1 and 4.2.

mmprotocoltrace command can be used on CES nodes to collect traces.

- Coordinate cluster wide traces
- Per client tracing options
- Network dumps
  - Captures packet data using dumpcap
- SMB tracing
  - Increase log level to debug 10
  - SMB system calls
- Object tracing
  - Increase log level for each openstack service

Tracing - mmprotocoltrace

#> mmprotocoltrace start network



```
Trace ID:    67b00fb7-2ca5-4be4-b90a-af1347931f6a
State:       ACTIVE
User ID:     root
Protocol:    network
Start Time:  16:13:52 03/11/2015
End Time:    16:23:52 03/11/2015
Client IPs:  []
Origin Node: bfn21-10g.gpfs.net
Nodes:
     Node Name:                 bfn22-10g.gpfs.net
     State:                     ACTIVE
     Trace Location:            /tmp/mmfs/network.20151103_161351.trc

     Node Name:                 bfn21-10g.gpfs.net
     State:                     ACTIVE
     Trace Location:            /tmp/mmfs/network.20151103_161351.trc
```

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

Grafana

Administrator want the ability to use 3rd party tools for monitoring their heterogenious environment

The Spectrum Scale performance monitoring framework will provide an interface for Grafana

- Grafana is the leading open source project for visualizing metrics
- Ability to show all Spectrum Scale performance metrics in Grafana
- openTSDB interface will translate requests into Spectrum Scale performance queries

Long Waiters

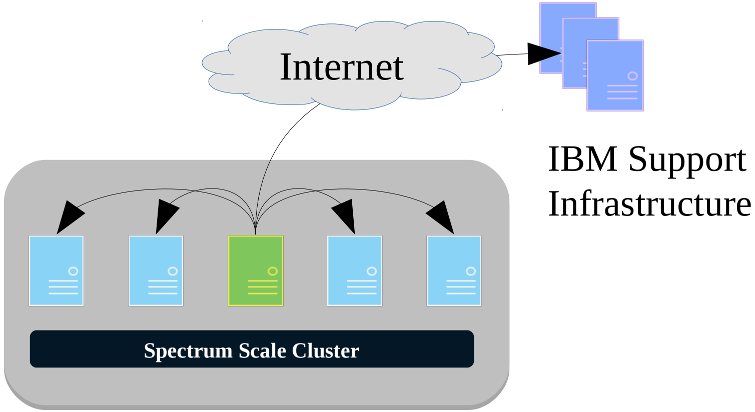
Looking at GPFS long waiters can help to understand the system load and find certain bottlenecks.

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

- Categorize waiters
  - e.g. 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 (e.g. throughput)

Callhome

Since 4.2 the system can automatically send problem reports to IBM



Callhome will help IBM to improve the product quality and to proactively react on problems

- Daily and weekly data collection
  - Spectrum scale configuration and logs
  - Statistical data (e.g. object statistics)
- Data collections triggered by events
  - e.g. SMB_down event will collect smb relevant data and send to IBM
- No customer data (e.g. file system content) is collected.
- Available for RHEL7 only
  - SLES 12 will be supported soon

and more

Additional improvements in all areas are planned to help clients with problem determination

- Top K processes
- Audit trail for configuration changing mmcommands
- Deadlock amelioration
- Automatic data collection on event
- Visualize heavy system jobs (e.g. restripe)
- OSCT consistency checking tool (Object)
- Best practice documentation

Problem Recovery

Automatic recovery

Automatic recovery of selected problems in the area of protocols has been added in 4.1.1 and 4.2

Certain problems can be recovered automatically without user intervention.

- Failover of CES IP addresses to recover from node failures
- Restart Protocol Services on error
  - Openstack services are restarted
  - Samba daemons will be restarted

DMPs

Directed Maintainance Procedures (DMPs) are used to guide the customer through the fixing procedure

For problems which can not be recovered automatically because customer interaction is needed:

- DMPs wizards in the UI
  - can help to gather the customer input
  - guide step by step through the fixing procedure

Are there common error scenario to add

?

Questions