



Problem Determination

Spectrum Scale 4.2.2 and outlook

Mathias Dietz



Mathias Dietz

IBM Research and Development in Mainz, Germany



- Spectrum Scale - Release Lead Architect
- Long experience with GPFS, protocols and system management
 - Current focus area: System Health & Problem Determination
 - Protocol integration in 4.1.1
 - Worked as architect for SONAS / V7KU



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.



System Health Monitoring

System Health Monitoring?



- **Why:** Detect common failure scenarios and guide users through the fix process.
- **How:** Uses callbacks, internal APIs, mmpmon data and mm-commands to monitor the component health
 - Dedicated monitoring daemon - independent from core mmfsd
 - Runs on all cluster nodes ($\geq 4.2.1$) / part of GPFS base package
 - Full Linux support on x86/pSeries/zLinux and partial AIX support (no Windows support)
- **What:** New mmhealth command (4.2.1) - Provide a central view of the system state with well-defined error events and states.
 - Drill down to find defective components
 - Shows current reasons of the found problems
 - User actions for fixing problems



Monitored services



- Over 500 events! Over 150 checks!

GPFS

- quorumloss, ccr_paxos_12_fail
- gpfs_down, longwaiters_found

PERFMON

- pmsensors_up /-down
- pm_collector_up /-down

DISK

- disk_found /-vanished
- disk_up /-down

FILESYSTEM

- fserrinodenummismatch, stale_mount
- fserrallocblock, fserrbaddirblock

CLOUDGW

- cloudgateway_down
- cloudgw_restart

HADOOPCON

- hadoop_namenode_up /-down
- hadoop_datanode_up /-down

NETWORK

- network_down, network_ips_down
- bond_down, network_link_down

GUI

- gui_up /-down
- gui_warn



Monitored services



- More components added in 4.2.2
- Existing component monitoring has been extended

OBJECT	object authentication AUTH_OBJ	Hadoop connector HADOOPCON	transparent cloud tiering (TCT) CLOUDGW
NFS	file authentication AUTH	FILESYSTEM	DISK
SMB	CES-relevant networks CESNETWORK	performance monitor PERFMON	GUI
block level storage BLOCK	GPFS-relevant networks NETWORK	common events GPFS	REST API monitoring SCALEMGMT
GNR enclosure ENCLOSURE	GNR physical disk PHYSICALDISK	CSM-relevant events CLUSTERSTATE	New in 4.2.2
GNR array ARRAY	GNR virtual disk VIRTUALDISK	GNR recovery group RECOVERYGROUP	



Extension details (ESS only)



- GNR (Native RAID):
 - Recovery Groups
 - Declustered Arrays
 - Virtual Disks
 - Physical Disks
 - Enclosures
 - DCM, ESM
 - Temp & Voltage Sensors
 - Fans
 - Power Supplies
 - Firmware Levels
 - Enclosure Firmware
 - Drive Firmware
 - Adapter Firmware
 - Adapter Bios

```
[root@gssio1 ~]# mmhealth node show -v
Node name:          gssio1-hs.gpfs.net
Node status:       HEALTHY

Component          Status              Reasons
-----
GPFS                HEALTHY             -
FILESYSTEM
  Basic1            DEGRADED           stale_mount, stale_mount, stale_mount
  Custom1          HEALTHY            stale_mount
GNR                 DEGRADED           gnr_array_needservice, enclosure_needs..
ARRAY              DEGRADED           gnr_array_needservice
  rg_gssio1-hs/DA1 DEGRADED           gnr_array_needservice
  rg_gssio1-hs/NVR HEALTHY             -
  rg_gssio1-hs/SSD HEALTHY             -
  rg_gssio2-hs/DA1 HEALTHY             -
  rg_gssio2-hs/NVR HEALTHY             -
  rg_gssio2-hs/SSD HEALTHY             -
ENCLOSURE          DEGRADED           enclosure_needservice
  SV52122944       DEGRADED           enclosure_needservice
  SV53058375       HEALTHY             -
PHYSICALDISK      DEGRADED           gnr_pdisk_replaceable, gnr_pdisk_re..
  rg_gssio1-hs/e1d1s01 HEALTHY             -
  rg_gssio1-hs/e2d5s03 FAILED             gnr_pdisk_replaceable
  rg_gssio2-hs/e1d1s01 FAILED             gnr_pdisk_replaceable
RECOVERYGROUP     HEALTHY             -
  rg_gssio1-hs     HEALTHY             -
  rg_gssio2-hs     HEALTHY             -
VIRTUALDISK       HEALTHY             -
  rg_gssio1_hs_Basic1_data_0HEALTHY -
  rg_gssio1_hs_Basic1_system_0HEALTHY -
  rg_gssio1_hs_loghome HEALTHY             -
  rg_gssio1_hs_logtip HEALTHY             -
  rg_gssio1_hs_logtipbackupHEALTHY -
  rg_gssio2_hs_Basic1_data_0HEALTHY -
  rg_gssio2_hs_Basic1_system_0HEALTHY -
...
```


Monitoring details



- **TCT**
 - 5 events in 4.2.1 → 47 events in 4.2.2
 - Monitoring service, account and filesystem
- **DISK**
 - Increased performance and stability
 - Reduced system load
- **GUI**
 - Integrated GUI's own monitoring data
- **REST API**
 - Basic Monitoring (http request)



Mmhealth command and GUI integration

Improvements to mmhealth



New in 4.2.2

- Listing cluster health overview
- Show date/time of last state change
- Show detail information about an event
- Improved Performance - reduced impact on system load in 4.2.1
- GUI - Health Overview page
- GUI - Finer granularity for Notification configuration



Listing cluster health overview



- The cluster overview
 - Shows cluster related events / incl. node connectivity checks (heartbeat)

```
mmhealth cluster show <component_name>
```

- Is my cluster working fine?

```
[root@ch-41 ~]# mmhealth cluster show
```

Component	Total	Failed	Degraded	Healthy	Other
NODE	5	0	1	4	0
GPFS	5	0	0	5	0
NETWORK	5	0	0	5	0
FILESYSTEM	1	0	0	1	0
DISK	2	0	0	2	0
CES	2	1	0	1	0
PERFMON	3	0	0	3	0





- Where/what is the CES problem?

```
[root@ch-41 ~]# mmhealth cluster show ces
```

Component	Node	Status	Reasons
CES	ch-41.localnet.com	HEALTHY	-
CES	ch-42.localnet.com	FAILED	ces_network_ips_down, nfs_in_grace, nfsd_down



Show date/time of last state change



- Time context for the last state change
 - human and machine-readable: at your choice!
- When did it happen?

```
[root@ch-41 ~]# mmhealth node show -N ch-42
```

```
Node name:      ch-42.localnet.com
Node status:    DEGRADED
Status Change: 10 min. ago
```

Component	Status	Status Change	Reasons
GPFS	HEALTHY	7 days ago	-
NETWORK	HEALTHY	8 days ago	-
FILESYSTEM	HEALTHY	7 days ago	-
DISK	HEALTHY	7 days ago	-
CES	FAILED	9 min. ago	nfsd_down, ces_network_ips_down, nfs_in_grace
PERFMON	HEALTHY	8 days ago	-



Show detail information about a event



- Instant help
 - `mmhealth event show <event_name>`
- What is `nfsd_down` and how do I fix it?

```
[root@ch-41 ~]# mmhealth event show nfsd_down
Event Name:      nfsd_down
Event ID:        999167
Description:     Checks for a NFS service process
Cause:           The NFS server process was not detected
User Action:     Check the health state of the NFS server and restart, if
necessary. The process might hang or is in a defunct state
Severity:        ERROR
State:           FAILED
```



Follow user action to fix the problem



- Let us fix the problem:

```
[root@ch-41 ~]# ssh ch-42
Last login: Wed Nov  9 11:09:42 2016 from ch-41.localnet.com
[root@ch-42 ~]# systemctl status nfs-ganesha
nfs-ganesha.service - NFS-Ganesha file server
   Loaded: loaded (/usr/lib/systemd/system/nfs-ganesha.service; disabled)
   Active: inactive (dead) since Wed 2016-11-09 11:10:31 CET; 14min ago
     Docs: http://github.com/nfs-ganesha/nfs-ganesha/wiki
   Process: 15220 ExecStop=/bin/dbus-send --system --dest=org.ganesha.nfsd
--type=method_call /org/ganesha/nfsd/admin org.ganesha.nfsd.admin.shutdown (code=exited,
status=0/SUCCESS)
   Main PID: 25484 (code=exited, status=0/SUCCESS)

Oct 31 15:56:14 ch-42.localnet.com systemd[1]: Starting NFS-Ganesha file server...
Oct 31 15:56:14 ch-42.localnet.com systemd[1]: Started NFS-Ganesha file server.
Nov 09 11:10:21 ch-42.localnet.com systemd[1]: Stopping NFS-Ganesha file server...
Nov 09 11:10:31 ch-42.localnet.com systemd[1]: Stopped NFS-Ganesha file server.
```



Problem solved



- Let us fix the problem:

```
[root@ch-42 ~]# systemctl start nfs-ganesha
[root@ch-42 ~]# mmhealth node show
```

```
Node name:      ch-42.localnet.com
Node status:    HEALTHY
Status Change: 3 min. ago
```

Component	Status	Status Change	Reasons
GPFS	HEALTHY	7 days ago	-
NETWORK	HEALTHY	8 days ago	-
FILESYSTEM	HEALTHY	7 days ago	-
DISK	HEALTHY	7 days ago	-
CES	HEALTHY	3 min. ago	-
PERFMON	HEALTHY	8 days ago	-



Problem solved



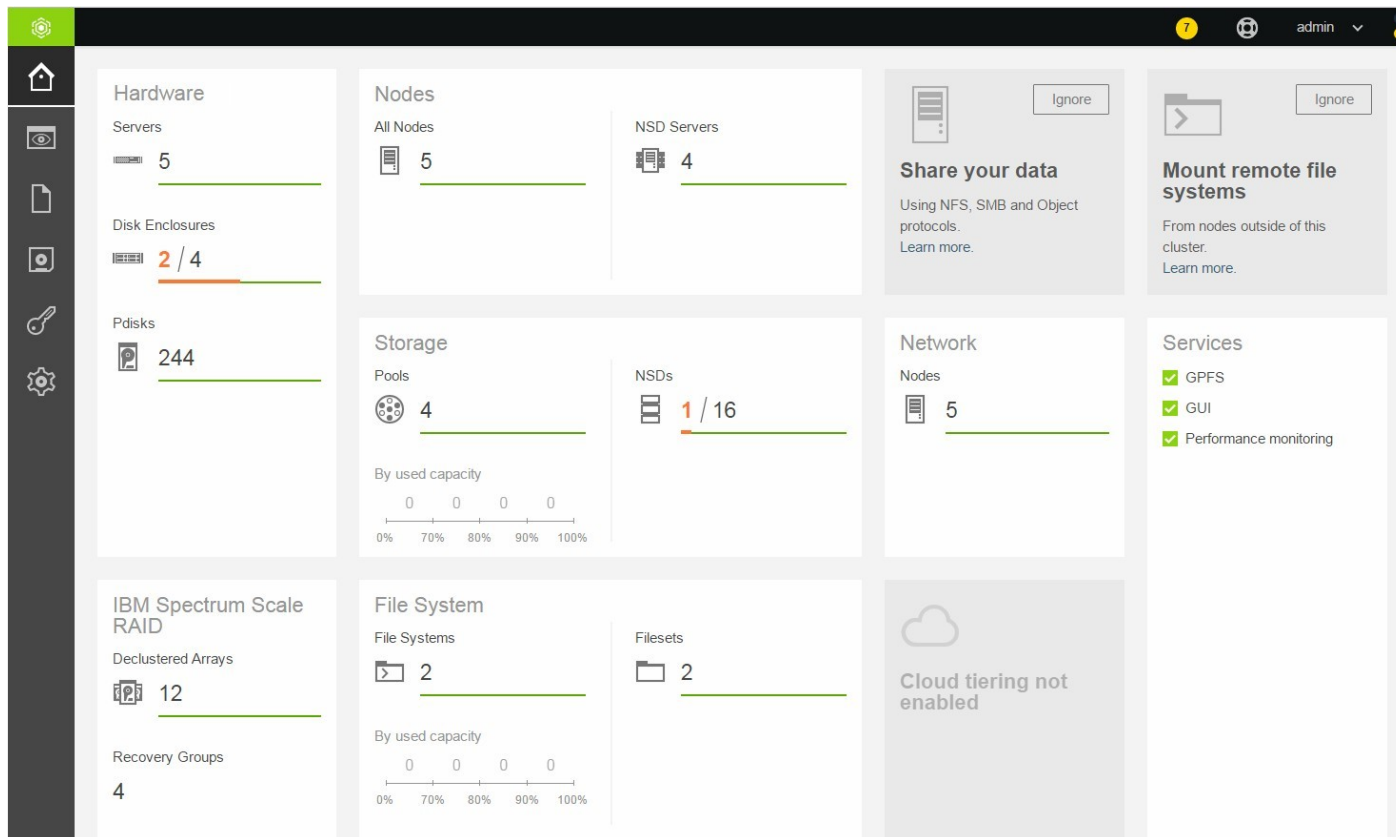
- What is with the cluster?

```
[root@ch-42 ~]# mmhealth cluster show
```

Component	Total	Failed	Degraded	Healthy	Other
-----	-----	-----	-----	-----	-----
NODE	5	0	0	5	0
GPFS	5	0	0	5	0
NETWORK	5	0	0	5	0
FILESYSTEM	1	0	0	1	0
DISK	2	0	0	2	0
CES	2	0	0	2	0
PERFMON	3	0	0	3	0



Health Overview page



Finer granularity for Notification configuration



Create Recipient [X]

Name:

Email address:

Severity

Events

- Declustered Array
- Block and iSCSI services
- CES Network
- Transparent Cloud Tiering
- Cluster State

Threshold (%)

- Create a list of recipients and define which events should trigger sending an email for this recipient
- Categories have been extended to match the categories in mmhealth in 4.2.2
- Events: send all events as they occur
- Report: generate a mail with all aggregated messages once per day
- Quota report: monitor usage of the system



Problem isolation / FTDC

First Time Data Capture



First Time Data Capture

Collect enough data to be able to debug a problem without asking for recreates

▪ **gpfs.snap tool**

- Cluster-wide snapshots with debugging relevant data / Easy to use and powerful
- Collecting Hadoop-relevant data
 - `gpfs.snap --hadoop`
 - Customizable to include user defined files
- Collecting Performance Monitor-relevant data

Details on the collected data: In the Spectrum Scale Documentation (“Using the gpfs.snap command”)



Protocol Tracing improvements in 4.2.2



mmprotocoltrace - automated tracing of protocol issues. Making a complex task quick and trivial

- SMB tracing already in 4.2.1
 - Did not help for AD-authentication related issues
- Added winbind tracing
 - Collecting isolated level 10 winbind logs from relevant nodes
 - Leaving system logs intact
 - Usage:

```
mmprotocoltrace start winbind  
... # recreating the issue  
mmprotocoltrace stop winbind
```



Isolate Network Problems



mmdiag command has been enhanced to show more details about the network connection

mmnetverify command can verify the network connectivity for a given list of nodes. It helps with isolating cluster networking issues

- Connect to other nodes through ssh
- Spawn netverify daemon on each node
- Verify port connectivity from any to any node
- Generate network traffic and evaluate network performance



Network Verification Tool



Command Syntax:

```
mmnetverify Operation [Operation...] [--N {Node[,Node...] | all}] [-a]
  [--target-nodes [Node,[Node...] | all]] [--configuration-file File] | --no-configuration]
  [--log-file File] [--min-bandwidth Number] [--verbose]
```

Supported Operations (4.2.2):

Local interface check, hostname resolution, ping, remote shell execution, remote copy, time sync, port checks (daemon, sdrserv, tscmd), network data transfer (small/medium/large size packets), bandwidth, flood



Isolate Active Directory Problems



With the help of mmadquery tool, users can verify that their authentication environment fulfills the requirements of Spectrum Scale and isolate Active Directory related problems.

Connectivity to Authentication servers

Find connectivity issues to domain controllers, e.g. caused by wrong firewall

Trust relationships

Verify multi-domain trust relationships

ID Mapping inconsistencies

Verify required UID/GID fields are filled and fit into configured id ranges to isolate the root cause of access failures.

The command has been introduced with SpectrumScale 4.2.1 and enhanced with 4.2.2





mmadquery Command Syntax

List AD Server objects

```
mmadquery list user|uids|gids|groups|dc|trusts|idranger
```

Check whether uids or gids are within locally defined id mapping range

```
mmadquery check uids|gids|idranger
```

Print number of users by group or number mapped and un-mapped user

```
mmadquery stats user|uids
```

Additions in 4.2.2

- *user and user groups for all and by domain*
- *number of users by user group and domain*
- *number of un-mapped user (user with no uidNumber)*
- *id ranges by domain*
- *user details (ids, primary group id....)*



Performance/Capacity Monitoring

Predefined filesystems capacity thresholds



Performance Monitoring:

>50 Performance sensors (GPFS IO, AFM, SMB, Object,NFS ...) and >1000 Metrics

Spectrum Scale 4.2.2 introduced **predefined filesystem capacity/inode thresholds.**

The capacity metrics will be frequently compared with the rules boundaries by internal monitor process. As soon as one of the metric values exceeds their threshold limit the system health daemon will receive an event notification from monitor process and generate log event and update filesystem status

The predefined filesystem capacity threshold limits break down to the following thresholds rules:

- Fileset-inode spaces
- Data pool capacity
- Metatadata pool capacity



Predefined filesystems capacity thresholds



```
># mmhealth node show -v
```

```
Node name:      node-11.nova.local  
Node status:    TIPS  
Status Change: 2017-03-07 22:18:54
```

Component	Status	Status Change	Reasons
GPFS	HEALTHY	2017-03-07 22:18:54	-
NETWORK	HEALTHY	2017-03-07 22:18:53	-
eth0	HEALTHY	2017-03-07 22:18:53	-
FILESYSTEM	DEGRADED	2017-03-07 23:04:37	inode_high_error(gpfs0/lowinode)
gpfs0	FAILED	2017-03-07 23:04:37	inode_high_error
objfs	HEALTHY	2017-03-07 22:27:51	-
DISK	HEALTHY	2017-03-07 22:19:06	-
disk1	CHECKING	2017-03-07 22:20:06	-
disk2	CHECKING	2017-03-07 22:27:37	-
GUI	HEALTHY	2017-03-07 22:23:20	-
PERFMON	HEALTHY	2017-03-07 22:20:40	-

Predefined filesystems capacity thresholds



To view the list of defined threshold rules on the system, issue this command:

mmhealth thresholds list

```
mmhealth thresholds list
```

The system displays output similar to this:

```
### Threshold Rules ###
```

```
-----  
- id: 001   designation: POOL-DATA   type: G metric: pool_data size usage   filterBy: perPool   level: HIGH_WARN   value: 80.0  
- id: 002   designation: POOL-DATA   type: G metric: pool_data size usage   filterBy: perPool   level: HIGH_ERROR  value: 90.0  
- id: 003   designation: POOL-METADATA type: G metric: pool_metadata size usage filterBy: perPool   level: HIGH_WARN   value: 80.0  
- id: 004   designation: POOL-METADATA type: G metric: pool_metadata size usage filterBy: perPool   level: HIGH_ERROR  value: 90.0  
- id: 005   designation: INODE       type: G metric: fileset_inode size usage filterBy: perFileset level: HIGH_WARN   value: 80.0  
- id: 006   designation: INODE       type: G metric: fileset_inode size usage filterBy: perFileset level: HIGH_ERROR  value: 90.0
```



Performance Monitoring Bridge for Grafana

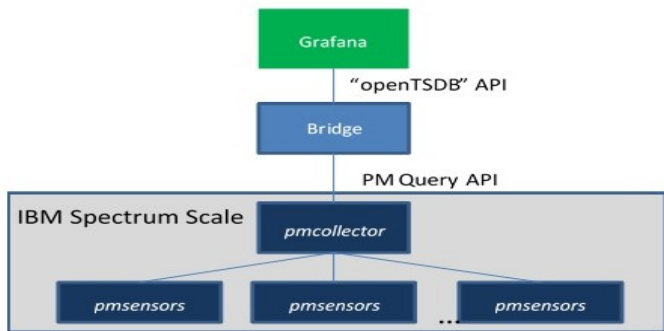


Figure 1. IBM Spectrum Scale integration framework for Grafana

Grafana

- an open source performance data graphical visualizer
- provides a powerful and elegant way to create, explore, and share dashboards and data with your team and the world.

IBM Spectrum Scale Performance Monitoring Bridge

- a Python application
- provides IBM Spectrum Scale performance data to Grafana in „openTSDB“ data exchange format
- communicates with active pmcollector via port 4242



Performance Monitoring Bridge for Grafana



1. Establish connection

2. Create Dashboard

The screenshot displays the Grafana interface. On the left, the 'Edit data source' dialog is open, showing the configuration for a data source named 'myBridge' of type 'OpenTSDB'. The 'Http settings' section shows the URL 'http://myBridge_ip_address:4242' and 'Access' set to 'direct'. The 'Opensdb settings' section shows 'Version' as '==2.2' and 'Resolution' as 'Second'. A green success message indicates 'Data source is working'. On the right, the main dashboard is visible, titled 'Main Dashboard'. It contains several panels: 'CPU usage', 'Memory usage', 'NETWORK', 'NSD Server Nodes', and 'Protocol Nodes'. Each panel displays a time-series graph with data points and a legend. The 'CPU usage' panel shows 'avg(cpu_user)'. The 'Memory usage' panel shows 'avg(mem_memtotal)', 'avg(mem_memfree)', 'avg(mem_buffers)', and 'avg(mem_cached)'. The 'NETWORK' panel shows 'sum(netdev_bytes_r)' and 'sum(netdev_bytes_s)'. The 'NSD Server Nodes' panel shows 'avg(gpfs_nsdds_read_ops)' and 'avg(gpfs_nsdds_write_ops)'. The 'Protocol Nodes' panel shows 'avg(gpfs_fs_read_ops)' and 'avg(gpfs_fs_write_ops)'. The dashboard also includes a 'DASHBOARD LIST' and 'CAPACITY VIEW Dashboards list' on the right side.



4.2.3 Release Outlook

Problem Determination Enhancements 2017



Further enhance problem detection / health monitoring in 4.2.3

- AFM cache monitoring
 - Runs on AFM gateway nodes
 - Monitors AFM Filesets → connection, queue drops, failover, recovery, etc

```
#> mmhealth node show afm
Node name:      afmGW-31.localnet.com
Component      Status          Status Change          Reasons
-----
AFM
  afmFs/afmCacheFset2  FAILED          Now
                       FAILED          21 days ago          afm_cache_unmounted(afmFs/afmCacheFset2)
                       FAILED          21 days ago          afm_cache_unmounted

Event          Parameter          Severity          Active Since          Event Message
-----
afm_cache_unmounted  afmFs/afmCacheFset2  ERROR          Now          Fileset afmFs/afmCacheFset2 is in
Unmounted state
```



Problem Determination Enhancements 2017



- Extended Network monitoring for Infiniband
 - Monitor Infiniband adapters (link state, port state)
 - Checks if RDMA is correctly configured and actually works (e.g. rdma libpath)
 - 17 new events: IB RDMA state and assist with problem solving

```
#> mmhealth node show network
```

```
Node name:      ifs6serv1.mainz.de.ibm.com
```

Component	Status	Status Change	Reasons
-----	-----	-----	-----
NETWORK	FAILED	Now	ib_rdma_libs_wrong_path, ib_rdma_nic_unrecognized(mthca0)
eno3	HEALTHY	1 day ago	-
mthca0	FAILED	Now	ib_rdma_nic_unrecognized

Event	Parameter	Severity	Active Since	Event Message
-----	-----	-----	-----	-----
ib_rdma_libs_wrong_path	NETWORK	ERROR	Now	The library files could not be found
ib_rdma_nic_unrecognized	mthca0	ERROR	Now	IB RDMA NIC mthca0 was not recognized



Problem Determination Enhancements 2017



Tips/Recommendations

- Inform customers about “potential” issues in the system
 - Warn users about common mis-configurations and non-optimal settings
 - e.g. pagepool is too small, performance sensors configuration wrong, etc
 - Introduce a new type of mmhealth events -> TIPS
 - Provide the ability to acknowledge/ignore a tip

```
# mmhealth node show gpfs
Node name:      node-11.novalocal

Component      Status      Status Change      Reasons
-----
GPFS           TIPS       3 days ago        gpfs_pagepool_small

Event           Parameter    Severity    Active Since    Event Message
-----
gpfs_pagepool_small  GPFS        TIP         3 days ago     The GPFS pagepool seems too small
```



Problem Determination Enhancements 2017



Tips in the GUI

The screenshot shows a web application interface for managing 'Tips'. The top navigation bar includes a home icon, a search icon, a refresh icon, a notification bell with '19' alerts, and a user profile 'admin'. The main content area is titled 'Tips' and features a table with columns: Current, Tip Name, Event Time, Message, Component, and Entity Name. A context menu is open over the second row, offering options: Filter by Date, Show entries within..., Reset Date Filter, Hide, Show, and Properties.

Current	Tip Name	Event Time	Message	Component	Entity Name
	gpfs_pagepool_small	03.03.17 09:18:27	The GPFS pagepool seems too small	GPFS	node-12.novalocal
	gpfs_pagepool_small	03.03.17 09:18:40	The GPFS pagepool seems too small	GPFS	node-13.novalocal
	all	03.03.17 10:26:15	The GPFS pagepool seems too small	GPFS	node-11.novalocal





Enhancements to the System Health framework

- Allow some control on health monitoring frequency
 - Monitoring interval High, Medium, Low
 - Trade-off: Failure detection time vs resource consumption
 - Examples:
 - Low = run monitors rarely → less overhead but longer failure detection time
 - High = run monitors very often → higher overhead, quick failure detection time

Command:

```
mmhealth config interval off | low | medium | high
```





Customer defined thresholds

- Allow customers to define thresholds on any performance metric
 - e.g. average network latency > 200ms
 - Ability to specify / configure warning and error levels
 - Set thresholds through command line (later through GUI)
 - Events will show up in mmhealth and GUI

Command:

```
mmhealth thresholds add { metric[:sum|avg|min|max|rate]|measurement [--errorlevel{threshold error limit} [--warnlevel{threshold warn limit}]|--direction {high|low}} [--sensitivity {bucketsize}] [--hysteresis {percentage}] [--filterBy] [--groupBy] [--name {ruleName}] [--errormsg {user defined action description}] [--warnmsg {user defined action description}]
```



Problem Determination Enhancements 2017



- New pre-defined thresholds for memory usage
 - memory free <50MB leads to error event, <100MB to warning event

```
[root@gpfsGUI-11 ~]# mmhealth thresholds list
### Threshold Rules ###
rule_name      metric          error  warn  direction  filterBy  groupBy  sensitivity
-----
InodeCapUtil_Rule  Fileset_inode  90.0  80.0  high      gpfs_cluster_name, gpfs_fs_name, gpfs_fset_name  300
DataCapUtil_Rule  DataPool_capUtil  90.0  80.0  high      gpfs_cluster_name, gpfs_fs_name, gpfs_diskpool_name  300
MemFree_Rule      mem_memfree     50000 100000 low     node  300
MetaDataCapUtil_Rule  MetaDataPool_capUtil  90.0  80.0  high      gpfs_cluster_name, gpfs_fs_name, gpfs_diskpool_name  300
[root@gpfsGUI-11 ~]#
```

- Rules can be deleted (or changed by adding them with different options)

Command:

```
mmhealth threshold delete { RuleName | All }
```



Problem Determination Enhancements 2017



REST API for System health state

- Query component state and events
- "https://localhost:443/scalemgmt/v2/nodes/ak-52/health/states?filter=state!=HEALTHY"

```
{ "states" : [ {  
  "activeSince" : "2017-03-07 13:46:06,370",  
  "component" : "PERFMON",  
  "entityName" : "ak-52.localnet.com",  
  "entityType" : "NODE",  
  "oid" : 175,  
  "reportingNode" : "ak-52.localnet.com",  
  "state" : "FAILED"  
}, {  
  "activeSince" : "2017-03-07 13:46:06,375",  
  "component" : "NODE",  
  "entityName" : "ak-52.localnet.com",  
  "entityType" : "NODE",  
  "oid" : 176,  
  "reportingNode" : "ak-52.localnet.com",  
  "state" : "DEGRADED"  
}],  
"status" : {  
  "code" : 200,  
  "message" : "The request finished successfully"  
}}
```



Problem Determination Enhancements 2017



More.....

- Improvements to mmnetverify tool to check protocol ports, bandwidth and better performance
- Better diagnostics of network issues inside the GPFS daemon
- FTDC improvements



Outlook

2H 2017 and beyond



Further enhance problem detection / health monitoring

- Extend existing monitoring to detect more problems, add additional Events and Tips !
 - GPFS Memory monitoring
 - Monitor critical threads (core daemon)
 - Extend network monitor, leverage mmnetverify command
 - Time sync checks
 - Zimon monitor, add connection check
- 500+ events, sometimes need more details and improved error recovery description
 - Define more detailed user actions
 - Add more DMPs (directed maintenance procedures)
 - Best practice guides



Problem Determination Enhancements



- Problem isolation
 - Extend mmnetverify for infiniband
 - Improve means to isolate protocol issues
 - Improve install toolkit error output for better usability
 - Improve FTDC
 - Extend gpfs.snap to collect additional data
 - Avoid gpfs.snap hangs on broken clusters
 - Investigation: Improve data collections for daemon crashes

- Improve Callhome
 - Upload snap data to existing PMR's
 - Enable Callhome at installation time (opt-out)
 - Semi automatic PMR creation (PMR opening through UI by User)



Problem Determination Enhancements



- Performance monitoring
 - Improve usability/robustness of mmpfmon command
 - Improve performance monitoring
 - installation and setup of performance sensors
 - Improve scalability of performance queries (multi-threading)
 - Update Grafana bridge to support latest Grafana releases
 - Externalize Top K process list
- GUI PD enhancements
 - Manage user defined thresholds
 - Performance monitoring (Zimon) configuration





Questions ?