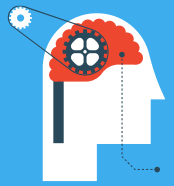


Spectrum Scale Expert Talks

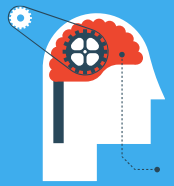
Episode 13:

Event driven data management and security using Spectrum Scale Clustered Watch Folder and File Audit Logging

IBM Spectrum Scale

Show notes:
www.spectrumscaleug.org/experttalks

Join our conversation:
www.spectrumscaleug.org/join



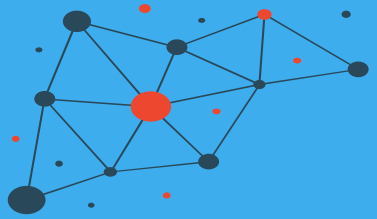
SSUG::Digital

Welcome to digital events!

IBM Spectrum Scale

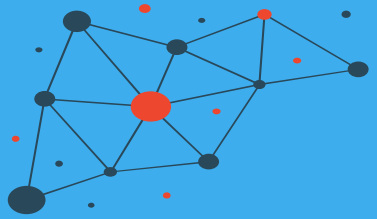
Show notes:
www.spectrumscaleug.org/experttalks

Join our conversation:
www.spectrumscaleug.org/join



Speaker

- Jürgen Hannappel (DESY)
- John Olson (USA)
- Luis Teran (USA)
- Jake Tick (USA)
- Simon Thomson (UG Host)



Check https://www.spectrumscaleug.org/experttalks for charts, show notes and upcoming talks

- Past talks:
  - 001: What is new in Spectrum Scale 5.0.5?
  - 002: Best practices for building a stretched cluster
  - 003: Strategy update
  - 004: Update on performance enhancements in Spectrum Scale (file create, MMAP, direct IO, ESS 5000)
  - 005: Update on functional enhancements in Spectrum Scale (inode management, vCPU scaling, NUMA considerations)
  - 006: Persistent Storage for Kubernetes and OpenShift environments
  - 007: Manage the lifecycle of your files using the policy engine
  - 008: Multi-node scaling of AI workloads using Nvidia DGX, OpenShift and Spectrum Scale
  - 009: Continental: Deep Thought – An AI Project for Autonomous Driving Development
  - 010: Data Accelerator for Analytics and AI (DAAA)
  - 011: What is new in Spectrum Scale 5.1.0?
- Today:
  - Dec 4: Lenovo: Spectrum Scale and NVMe storage
  - Dec 4: Event driven data management and security using Spectrum Scale Clustered Watch Folder and File Audit Logging

Use of the policy engine in data management.

J. Hannappel, S. Dietrich, M. Gasthuber

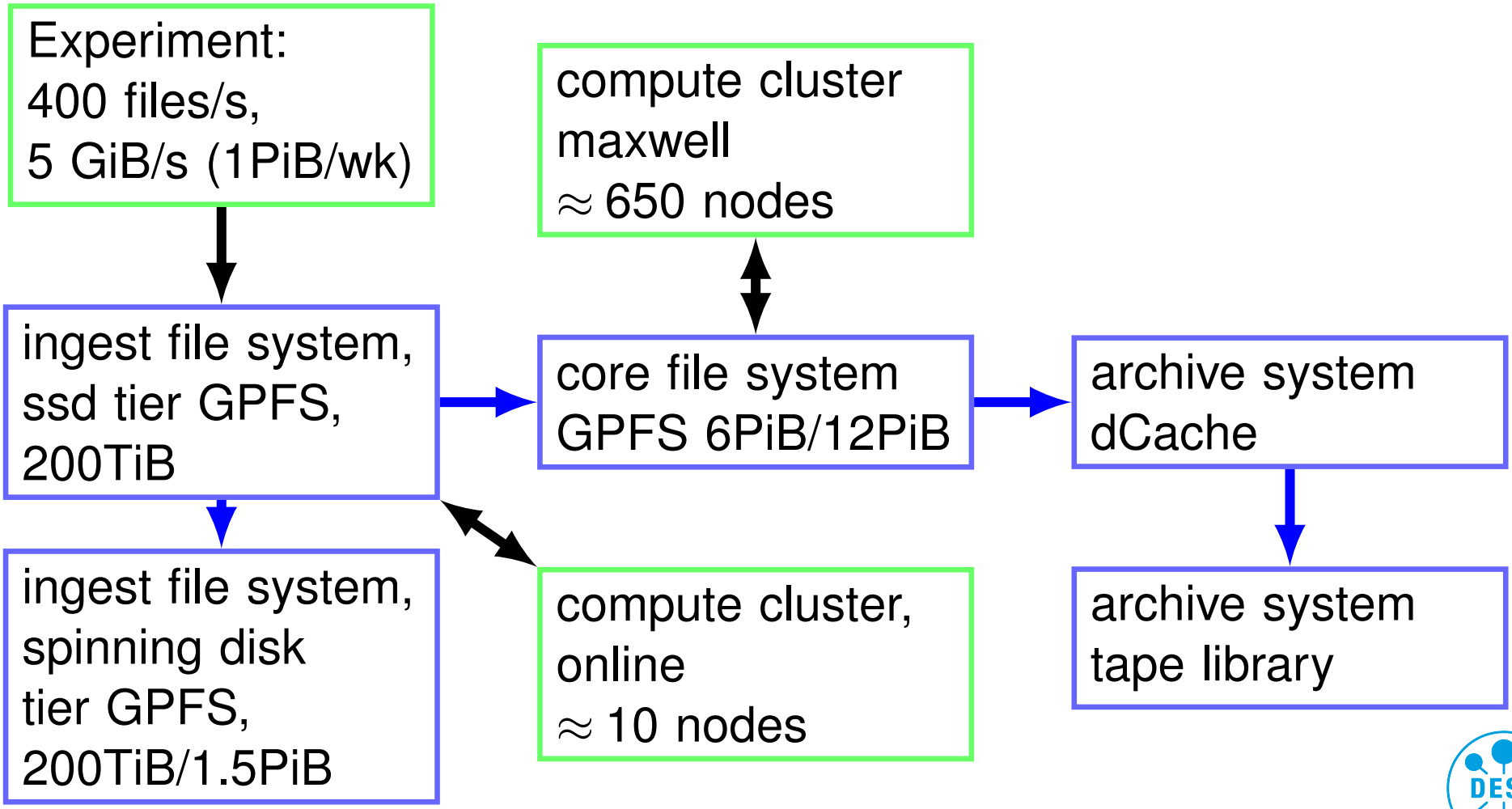
December 4, 2020

DESY and EuXFEL

- Desy
  - since 1959
  - Accelerator Science, Particle physics, Photon Science, Astro Particle
  - Hamburg and Zeuthen
  - $\approx$ 2300 employees, $\approx$ 3000 guest scientists
- EuXFEL
  - founded 2009, user operation 2017
  - Photon science
  - Schenefeld (near Hamburg)
  - 3.5 km long accel from Desy-Campus to EuXFEL

Setup

Similar setup for Petra III and EuXFEL



Data management tasks

Data sets organized in filesets

- Migrate data from ssd pool to disk pool
- Copy data from ingest to core fs
- Copy data from core fs to archive
- Generate file lists for user portal
- Detect unused data for cleanup

Data life cycle

Data sets organized in filesets

- Creation of Data: In ingest filesystem
- Transfer to core filesystem — Policy-assisted
- Copy to Archive — Policy-assisted
- Removal from ingest filesystem
- When data no longer accessed fo some time: — Policy-assisted
  do ΔCopy to Archive
  remove from core filesystem

Standard policy engine use

Task “Migrate data from ssd pool to disk pool”:

```
RULE 'InitialPlacement'
  SET POOL 'cache'
    LIMIT(75)

RULE 'FallbackPlacement'
  SET POOL 'data'

RULE 'PoolMigration'
  MIGRATE
    FROM POOL 'cache'
    THRESHOLD(60,50,0)
  TO POOL 'data'
```

Standard policy engine use

Task “Migrate data from ssd pool to disk pool”:

```
define(access_age_in_minutes,
   (INTEGER(((CURRENT_TIMESTAMP
              - ACCESS_TIME)SECONDS)) / 60.0))

RULE 'PoolMigration'
  MIGRATE
    FROM POOL 'cache'
    THRESHOLD(60,50,0)
    WEIGHT (access_age_in_minutes)
  TO POOL 'data'
    WHERE access_age_in_minutes > 2
```

via callback on lowDiskSpace event

List execution policy engine use

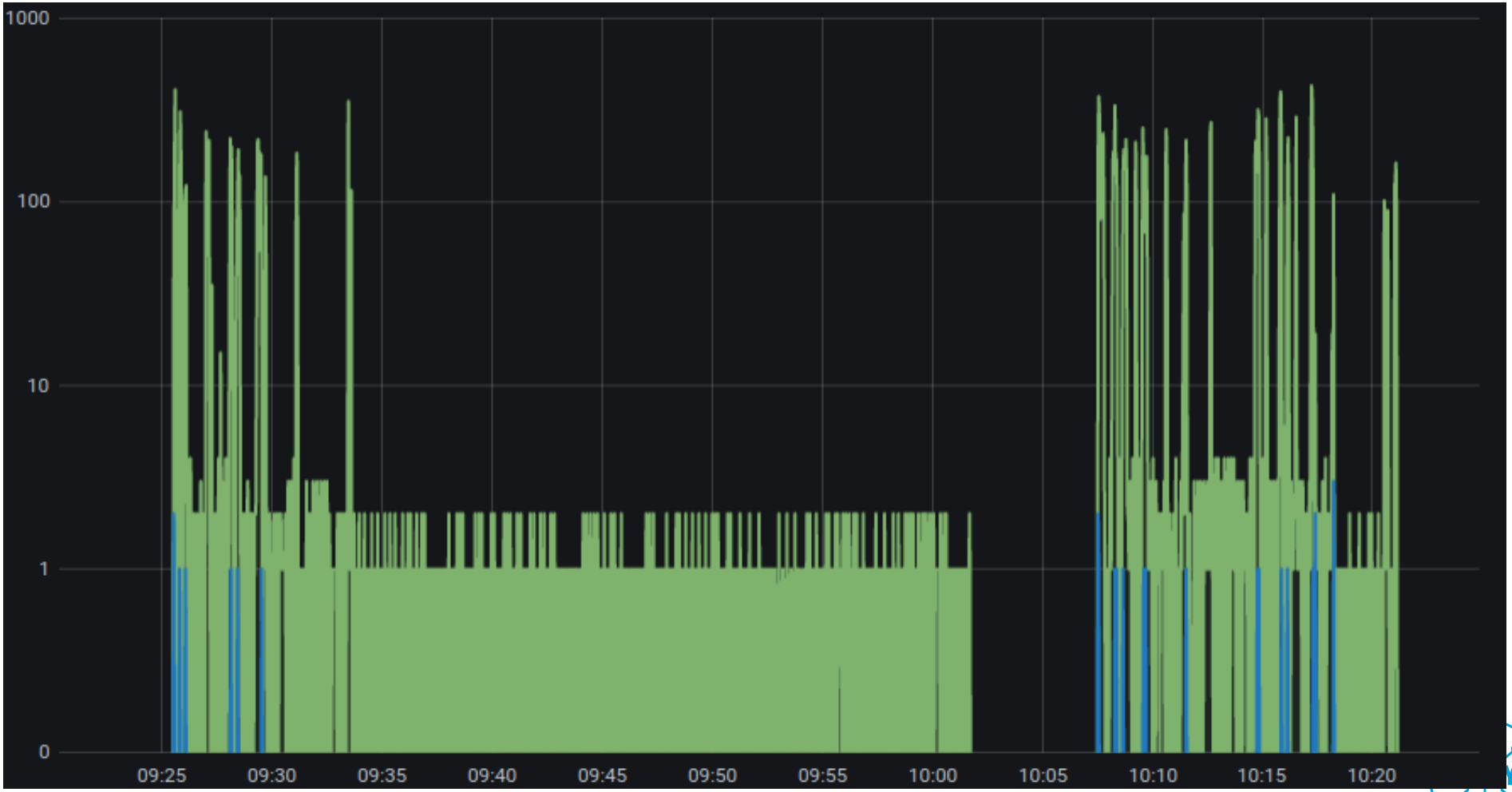
mmapplypolicy via crontab for copying data:

```
RULE
  EXTERNAL LIST 'BeamlineMigrator'
    EXEC '@CMAKE_INSTALL_PREFIX@/bin/migrateBeamline'

RULE 'BeamlineMigrator'
  LIST 'BeamlineMigrator'
    WEIGHT(DIRECTORY_HASH)
    WHERE
      access_age_in_minutes > 8 AND
      (FILESET_NAME LIKE '%-bt-%' OR FILESET_NAME LIKE '%-co-%') AND
      (
        CASE
          WHEN DISPLAY_NULL(XATTR('user.migrated')) != 'true'
            THEN TRUE
          WHEN DISPLAY_NULL(XATTR('user.mtime')) != VARCHAR(MODIFICATION_TIME)
            THEN TRUE
          WHEN DISPLAY_NULL(XATTR('user.file_size')) != VARCHAR(FILE_SIZE)
            THEN TRUE
          ELSE
            FALSE
        END
      ) AND
      XATTR('user.truncated') IS NULL
```

Policy driven copy: speed

Copy $\approx$ 400 files/s (at 2.5MiB/file)



Policy driven copy: summary

- Works reliably
- Policy engine allows efficient choice using metadata like xattrs much better than find + scripts with getfattr
- Driven by cronjob:
  - bursty load on the system
  - long delay between file close and copy

  ↪ use event triggered mechanism

Event triggered copy

Fast and low-latency copy from ingest to core FS required:

- conjob based copy has high latency
- AFM can't be used due to ownership transform requirements
- trigger copy from inotify or watchfolder
- usually fast (few 10 mSec) from close to finished copy
- difficult user ideas sometimes cause failures
  ↪ policy driven job as fallback

Summary

- Automatize what can be automatized
- Policy engine simplifies and speeds up many tasks reliably
- For fast actions event triggered methods

Disclaimer

IBM Spectrum Scale

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 Spectrum Scale

Agenda

1) Overview and limitations of pre 5.1.0 Spectrum Scale with Kafka integrated message queue

2) Overview of the new and more lean architecture for File Audit Logging and Clustered Watch Folder in Spectrum Scale 5.1.0

3) How to upgrade to the new architecture

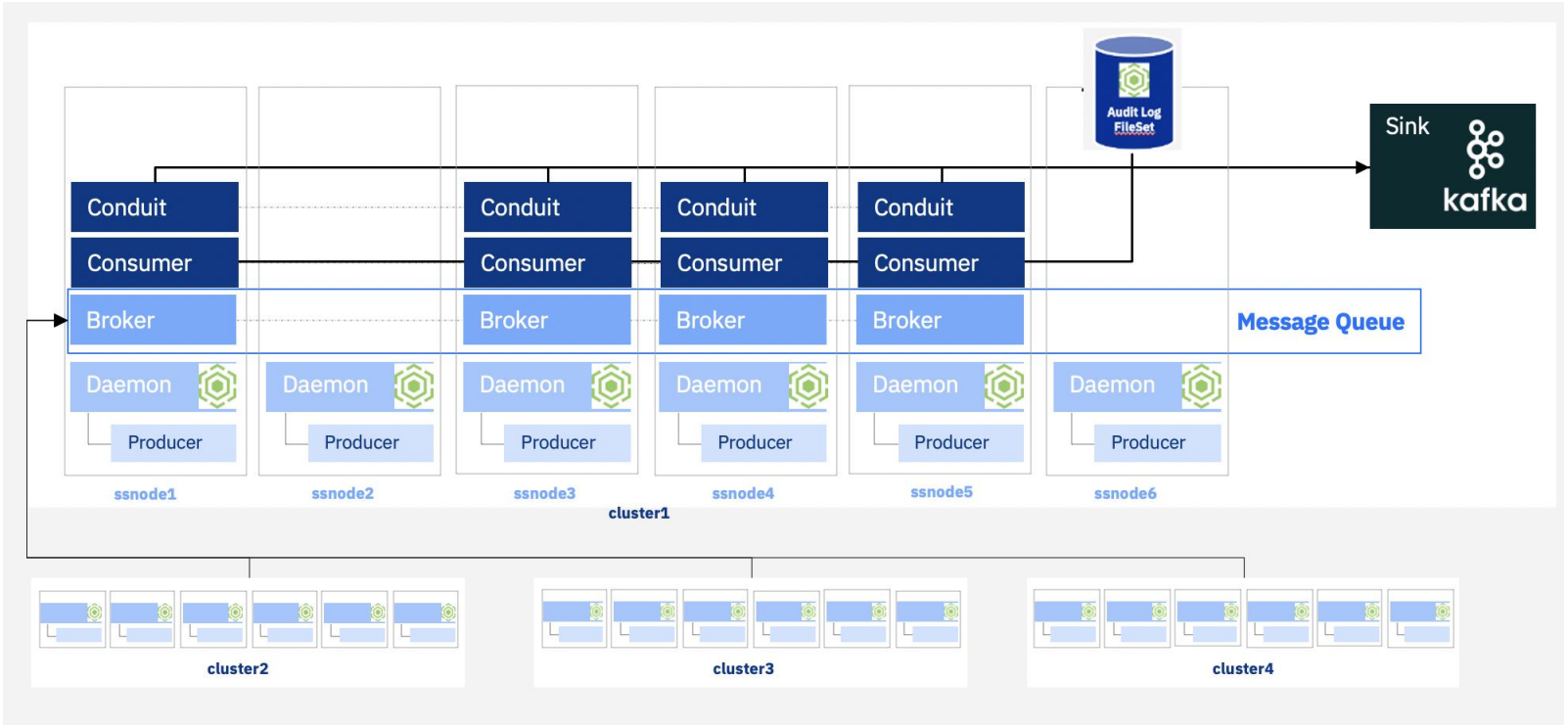
4) Use cases for File Audit Logging and Clustered Watch Folder

5) New additions to File Audit Logging - ACCESS DENIED

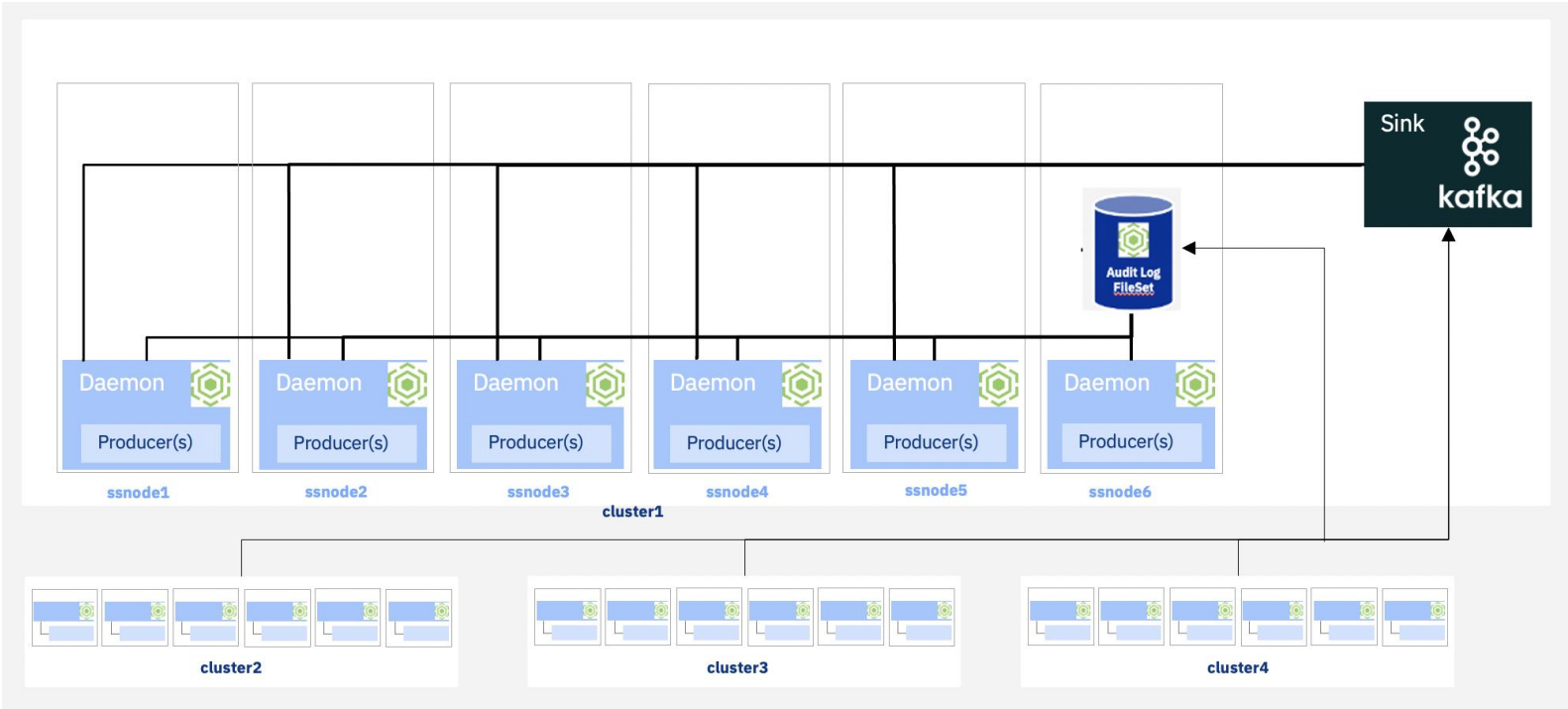
6) Best practices for File Audit Logging and Clustered Watch Folder

IBM

Overview of the Pre 5.1.0 Architecture



Overview of New 5.1.0+ Architecture



Benefits of New 5.1.0+ Architecture

- No brokers or zookeepers and no message queue
  - No requirement for a minimum of 3 Linux quorum nodes for zookeepers
  - No requirement for a minimum of 3 nodes for brokers
  - Less overhead because no extra Java processes are running on cluster nodes
  - No need for user space consumers to read from the message queue
- No more restrictions on stretch clusters
  - Broker-to-broker and zookeeper-to-zookeeper communications require short latencies for proper operation
  - Stretch clusters exacerbated communication problems when brokers and zookeepers split on both sides
  - No message queue means no need for brokers and zookeepers to communicate across both sides of stretch cluster
- No more local disk space required
  - Previously each filesystem audited or clustered watch enabled required a minimum of 20 GB free local disk space on each broker node
- All commands run much faster
  - No longer have to create/remove topics and no longer have to start/stop user space processes that would read from the message queue

Upgrading to the New Architecture

What was our goal for this procedure?

1. Make the transition to remove the message queue as seamless and painless as possible
2. Save state information so that any errors that are encountered can be remedied
   - Running the command again will start off at the last successfully completed step
3. Remove all traces of the message queue

What are the requirements and commands needed to upgrade?

1. All nodes in the cluster have to be at 5.1.0
   - minReleaseLevel 5.1.0.0 (mmchconfig release=LATEST / mmlsconfig release)
2. All filesystems with clustered watch or audit enabled must be upgraded to 5.1.0
   - FS version 24.00 (5.1.0.0) (mmchfs <device> -V full / mmlsfs <device> -V)
3. Run “mmmsgqueue config --remove-msgqueue”

Upgrading to the New Architecture (Continued)

What automated steps are taken while running the "mmmsgqueue config --remove-msgqueue" command?

1. Check that all filesystems associated with watches or audits are at filesystem version 5.1.0 or later
2. Save active clustered watch folder configuration
3. Disable active, auto disabled, suspended and resumed clustered watch folder instances
4. Save current file audit logging configuration
5. Disable file audit logging on enabled filesystems
6. Disable and remove message queue configuration
7. Reenable file audit logging on enabled filesystems, now using new policies
8. Reenable clustered watch folders that were previously active, now using new policies
9. For each Linux node in the cluster
    1. Remove gpfs.kafka package, if present
    2. Remove /opt/kafka/config contents and directory, if present
    3. Delete deprecated functionality

Use Case for File Audit Logging

IBM Spectrum Scale

An aerospace company is using Spectrum Scale for user accounts and classified files. The classified files are contained in 2 separate filesets: dont_look_here and Secret_Stuff, both in filesystem fs1.

All types of file access within the 2 classified filesets must be monitored, and the user who performed the operation as well as what operation was performed must be audited. Audit records must be kept for a minimum of 120 days.

To accomplish the above goals, the Spectrum Scale administrator runs a command similar to the following:

```
mmaudit fs1 enable --filesets dont_look_here,Secret_Stuff --retention
120
```

Use Case for File Audit Logging, Cont.

IBM Spectrum Scale

A banking company stores all financial information on a Spectrum Scale file system, fs1. This file system not only holds the financial information of the institution but it also holds temporary directories within 2 certain filesets that are used by data mining applications.

The company is legally required to audit the file system but does not want to have all of the temporary file activity within the 2 filesets (data_mine_process1 and data_mine_TMP) "flood" the audit logs.

To audit the entire file system except for the 2 filesets used by the data mining applications, the Spectrum Scale administrator runs a command similar to the following:

```
mmaudit fs1 enable --skip-filesets data_mine_process1,data_mine_TMP
```

Use Case for File Audit Logging, Cont.

IBM Spectrum Scale

A university has a storage cluster that no students or faculty can directly access. They also have a client cluster for each of the 4 main departments on campus that remote mounts their own filesystem, fs1, fs2, fs3, and fs4. Each department has multiple dependent and independent filesets they access on their filesystems. The Security Admin wants to be able to monitor each of the filesystems, each with their own requirements.

Since the storage cluster ”owns” the filesystems, the Security Admin can enable file audit logging on each of the filesystems with their own unique configs. The finance department needs to have a compliant audit fileset, watching for every event, and a retention policy of 2 years. The grounds keeping department just needs to know if anyone deleted any files from their databases and if anyone has been denied write access to their files.

Both of these use cases for auditing the finance and groundskeeping departments could be achieved with the following commands:

```
mmaudit fs1 enable --events ALL --compliant --retention 730

mmaudit fs2 enable --events UNLINK,ACCESS_DENIED
```

Use Case for Clustered Watch Folder

IBM Spectrum Scale

- Applications ingest data from sensors and write it to specific top tier filesets on a Spectrum Scale cluster. The admin has setup clustered watch folder to watch for the IN_CLOSE_WRITE event on those filesets. The admin has also written a consumer for the external kafka sink that pulls the file name and path from the event. That consumer process can take the path and the file name and move the files from the top tier storage to medium tier storage for a data scientist to analyze. This leaves the top tier storage open for more ingestion of data from the sensors.
- From the medium tier storage fileset, another watch can be setup to monitor when the data was accessed. Once we get the IN_CLOSE_WRITE or IN_CLOSE_NOWRITE event, the Kafka consumer process can then migrate that data off to tape.
- The admin configured a Kafka sink to run directly on the Spectrum Scale cluster. From two Spectrum Scale nodes, the admin runs custom Kafka consumer processes to execute the migration.

New Additions to File Audit Logging - ACCESS DENIED

Access Denied Overview

- An ACCESS_DENIED event is appended to the file audit log when a user attempts to access a file for an operation that the given user does not have the necessary permissions for (POSIX only).
- The ACCESS_DENIED event has the same fields as other events in file audit logging, with the addition of the accessMode field.
- The accessMode field specifies if the ACCESS_DENIED event is generated for a read, write, execute, control, delete, or insert operation.
- The ACCESS_DENIED event is generated whenever a file access operation is denied through the file system daemon.

How to update audited events

- Requires latest filesystem version (run *mmchfs <device> –V full* after upgrading to Spectrum Scale version 5.1.0.0)
- Verify which events are enabled with *mmaudit all list --events*
- *mmaudit <device> update --events { Event1[,Event2...] | ALL }*

Example of Audit Record

- {"LWE_JSON": "0.0.3", **"path": "/ibm/fs0/newfile2"**, "clusterName": "mycluster", "nodeName": "mynode", "nfsClientIp": "", "fsName": "fs0", **"event": "ACCESS_DENIED"**, "inode": "330889", "linkCount": "1", "openFlags": "0", "poolName": "system", "fileSize": "0", "ownerUserId": "0", "ownerGroupId": "0", "atime": "2020-11-06_10:46:36-0700", "ctime": "2020-11-06_10:46:36-0700", "mtime": "2020-11-06_10:46:36-0700", "eventTime": "2020-11-18_14:42:12-0700", **"clientUserId": "1001", "clientGroupId": "1001", "accessMode": "WRITE"**, **"processId": "15494", "permissions": "200100644"**, "acls": null, "xattrs": null, "subEvent": "NONE"}

Best Practices for File Audit Logging

- Eliminating Unwanted Noise
  - Enabling File Audit Logging on a filesystem means that for every audited operation, there needs to be an audit record generated. It is best practice to only audit events that are of interest (if possible)
    - Only enable audit for a subset of events that you are interested in
    - Only audit a subset of filesets
    - Skip noisy filesets that might not be of interest
- Have the Audit Fileset on faster tier storage (SSDs)
  - Having the File Audit Logs reside in faster tier storage allows for faster writes
    - Leads to less resource contention when buffering audit events
- Use mmdiag to validate producers on all nodes
  - Use mmdiag --eventproducers to look at statistics about the audit records being generated

Best Practices for Clustered Watch Folder

- Eliminating Unwanted Noise
  - Enabling Watch on a filesystem means that for every watched operation, there needs to be a watch event generated. It is best practice to only watch events that are of interest (if possible)
    - Only enable watch for a subset of events that you are interested in
    - Only watch filesystems/filesets/inodespaces that are of interest
- Make sure gpfs.librdkafka is installed on nodes from which events can be generated
- Make sure the external Kafka cluster infrastructure that Spectrum Scale nodes are delivering events to can handle the Spectrum Scale workload.
  - Look at mmdiag --eventproducer to validate events being generated and look at statistics about messages delivered to Kafka

Links and Resources

IBM Spectrum Scale

Clustered watch folder intro

File audit logging intro

Configuring clustered watch folder

Configuring file audit logging

Monitoring clustered watch folder

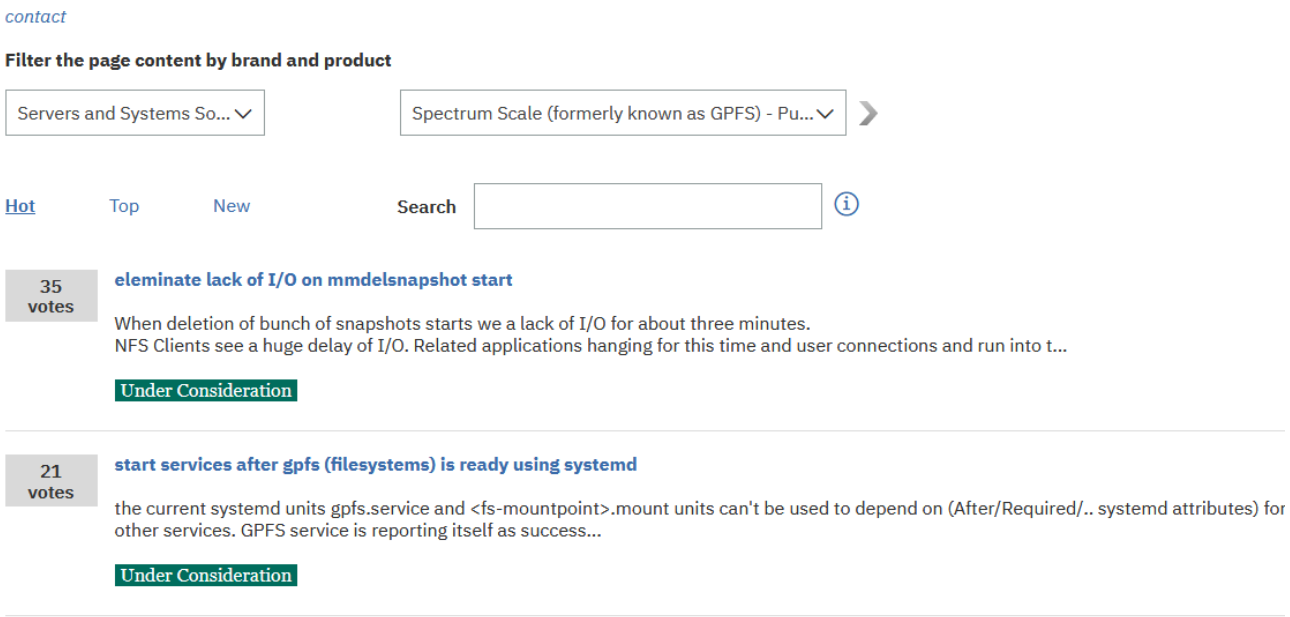
Monitoring file audit logging

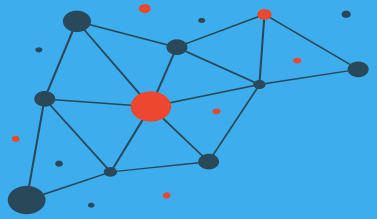
Upgrading both clustered watch folder and file audit logging

Log your RFE!

https://www.ibm.com/developerworks/rfe/execute?use_case=productsList

- Spectrum Scale (formerly known as GPFS) - Private RFEs
- Spectrum Scale (formerly known as GPFS) - Public RFEs





Check https://www.spectrumscaleug.org/experttalks for charts, show notes and upcoming talks

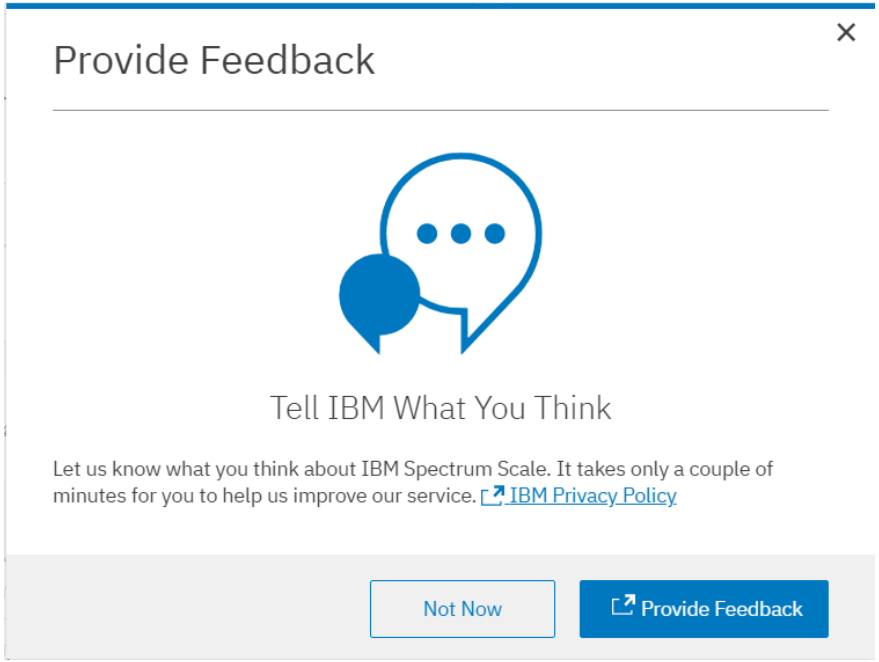
- Past talks:
  - 001: What is new in Spectrum Scale 5.0.5?
  - 002: Best practices for building a stretched cluster
  - 003: Strategy update
  - 004: Update on performance enhancements in Spectrum Scale (file create, MMAP, direct IO, ESS 5000)
  - 005: Update on functional enhancements in Spectrum Scale (inode management, vCPU scaling, NUMA considerations)
  - 006: Persistent Storage for Kubernetes and OpenShift environments
  - 007: Manage the lifecycle of your files using the policy engine
  - 008: Multi-node scaling of AI workloads using Nvidia DGX, OpenShift and Spectrum Scale
  - 009: Continental: Deep Thought – An AI Project for Autonomous Driving Development
  - 010: Data Accelerator for Analytics and AI (DAAA)
  - 011: What is new in Spectrum Scale 5.1.0?
- Today:
  - Dec 4: Lenovo: Spectrum Scale and NVMe storage
  - Dec 4: Event driven data management and security using Spectrum Scale Clustered Watch Folder and File Audit Logging

Thank you!

IBM Spectrum Scale

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



Spectrum Scale User Group

The Spectrum Scale (GPFS) User Group is free to join and open to all using, interested in using or integrating IBM Spectrum Scale.

The format of the group is as a web community with events held during the year, hosted by our members or by IBM.

See our web page for upcoming events and presentations of past events. Join our conversation via mail and Slack.

www.spectrumscaleug.org