XRootD Monitoring Flow

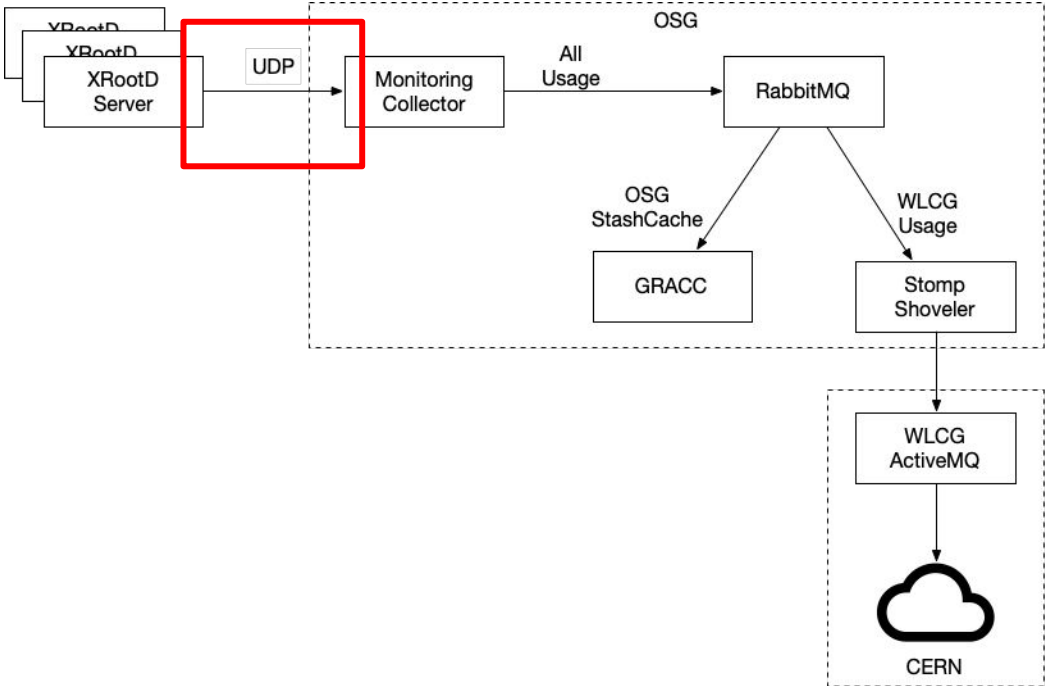
Derek Weitzel
Borja Garrido Bear

Motivation

- We want to be **confident** in our XRootD transfer accounting
- In OSG's first validation, we identified UDP Fragmentation as the largest culprit for missing validation data
- In OSG's second validation, we found that scaling the monitoring stream can cause some UDP packet loss.

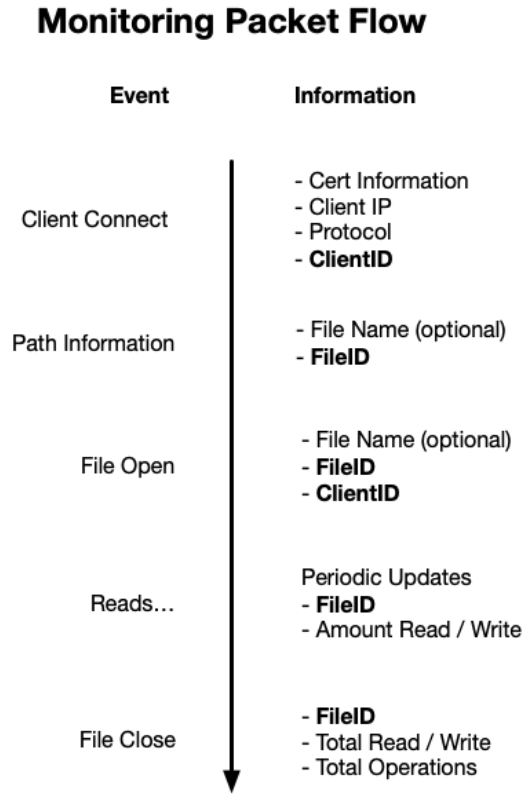
XRootD Monitoring = XRootD Detailed Monitoring

- Current monitoring uses detailed collector packets



Why XRootD Detailed Monitoring is Hard - Format

- Collector has to keep a lot of state
- Potential for packet loss means we have to place TTL on state
- Time between client connect and file close can be **hours**
- Must “join” different messages, but may lose packets
- For example, if you get a file close without the corresponding file open, then no idea what file was read.



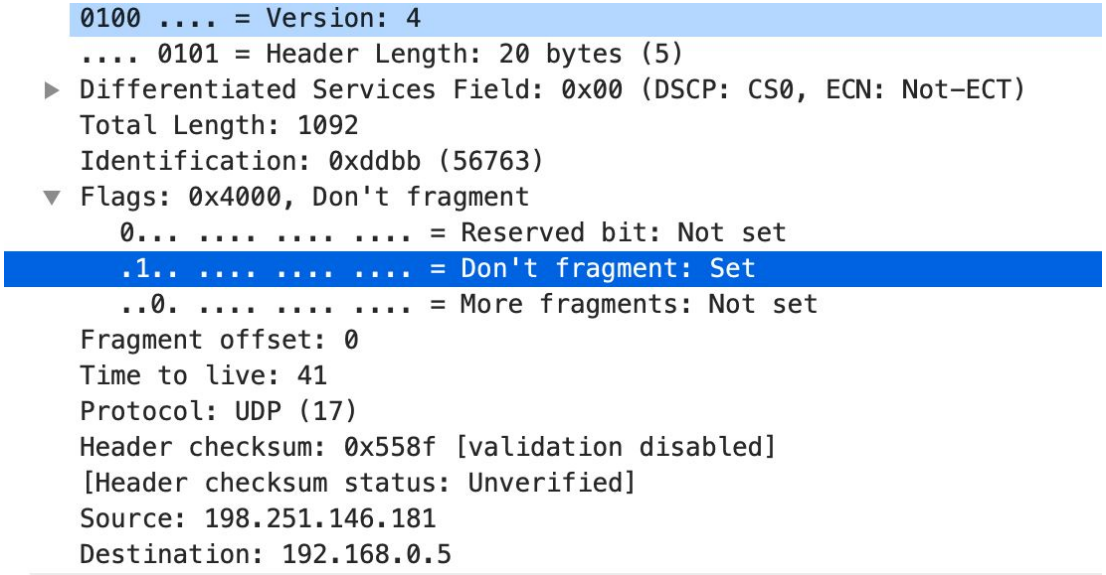
Observations from validation v1

- Small bugs in Collector
- Incorrect assumption: Sequence numbers in monitoring packets are not a reliable measure of missed packets (since fixed)
- **UDP fragmentation caused significant loss**

Report: https://doi.org/10.5281/zenodo.3981359

UDP Fragmentation

- UDP Fragmentation is a known problem: https://blog.cloudflare.com/ip-fragmentation-is-broken/
- The very Zoom meeting you are on uses UDP packets:



Tests performed in validation 2

In the second version of our validation we wanted to find out:

1. If sending monitoring data simultaneously from multiple XRootD servers would show any kind of data loss.
2. What is the maximum rate at which our collector can process monitoring records.

Monitoring data from multiple XRootD servers

On each test a client will request ‘N’ number of random files to each of the ‘M’ servers, then wait for a second and repeat until a total amount of ‘O’ files is reached where:

N - Req. rate
M - Num. Servers
O - Total files req.

After each test. we will pull the recorded data from rabbitMQ and compare with what we requested.

With this experiment we concluded that data loss due to scale is negligible

Num. Servers	Files req. per server	Total files req.	Req. rate	Files recorded avg.	Success %
2	100	200	20/s	200.00	100.00%
4	100	400	20/s	400.00	100.00%
8	100	800	20/s	800.00	100.00%
32	100	3,200	20/s	3196.67	99.90%
50	100	5,000	20/s	5000.00	100.00%
50	200	10,000	50/s	10000.00	100.00%
50	400	20,000	80/s	19992.33	99.96%
50	800	40,000	100/s	39991.00	99.98%

XRootD Monitoring - 2 components

- **Shoveler (simple):**
  - **Runs at Sites**
  - Collects the monitoring UDP packets from XRootD
  - “Packages” the UDP messages and sends them to a reliable message bus

- **Collector (complicated):**
  - **Runs Centrally**
  - Parses monitoring messages
  - Keeps state
  - Processes packets to extract VO, application info, type of transfer

Solution - XRootD Monitoring Shoveler

- Design and develop a “shoveler” from the UDP format to a resilient format (Message Bus)

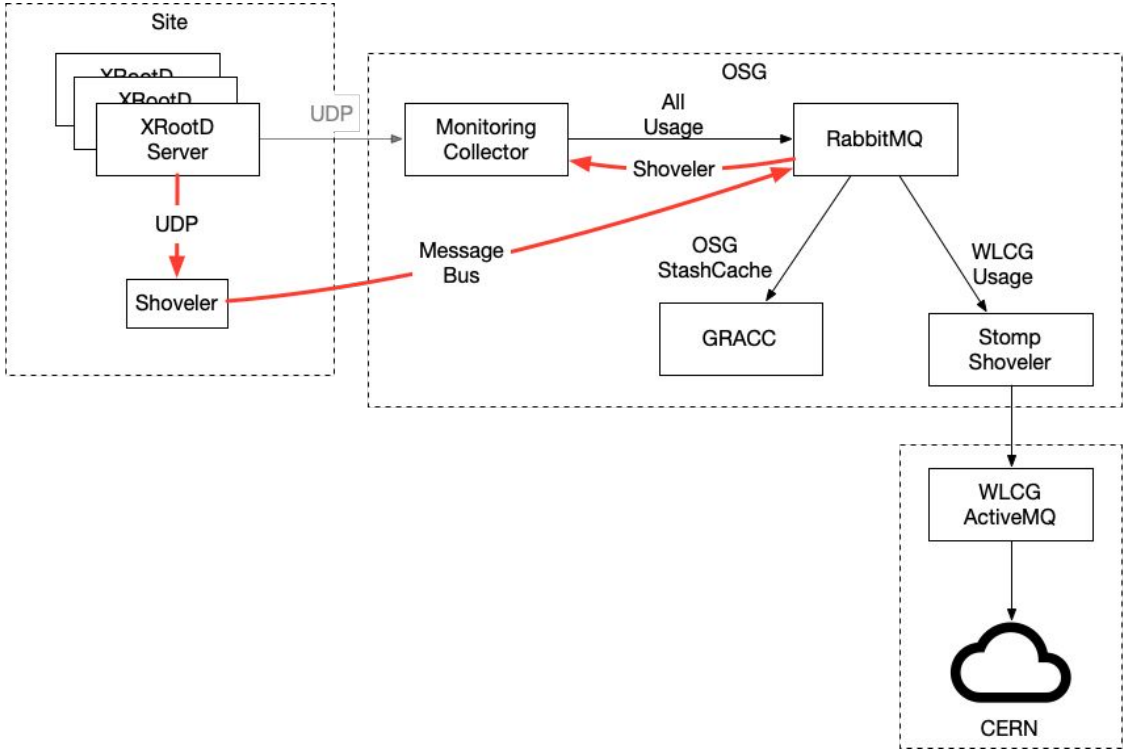
- The shoveler is simple, does no parsing or aggregation of records:

Shoveler Operation

1. Receives Packets
2. Very simple validation
3. Packages the data packet (base64’s the data, puts in json with other metadata)
4. Reliably sends to message bus

Shoveler

- A lightweight shoveler from UDP to a resilient transfer method
- Connection to RabbitMQ
- Already running at CMS Tier2's UCSD, Nebraska, & Florida.



Design Decisions

- The shoveler is purposefully “simple”
- The collector performs all stateful logic

- When shoveler is disconnected from message bus, it will write messages to disk and replay them when reconnected.
  - A production shoveler will write ~30MB of data a day to disk if disconnected.

Shoveler

Available at
https://github.com/opensciencegrid/xrootd-monitoring-shoveler/releases

Will be available in OSG's repos soon.

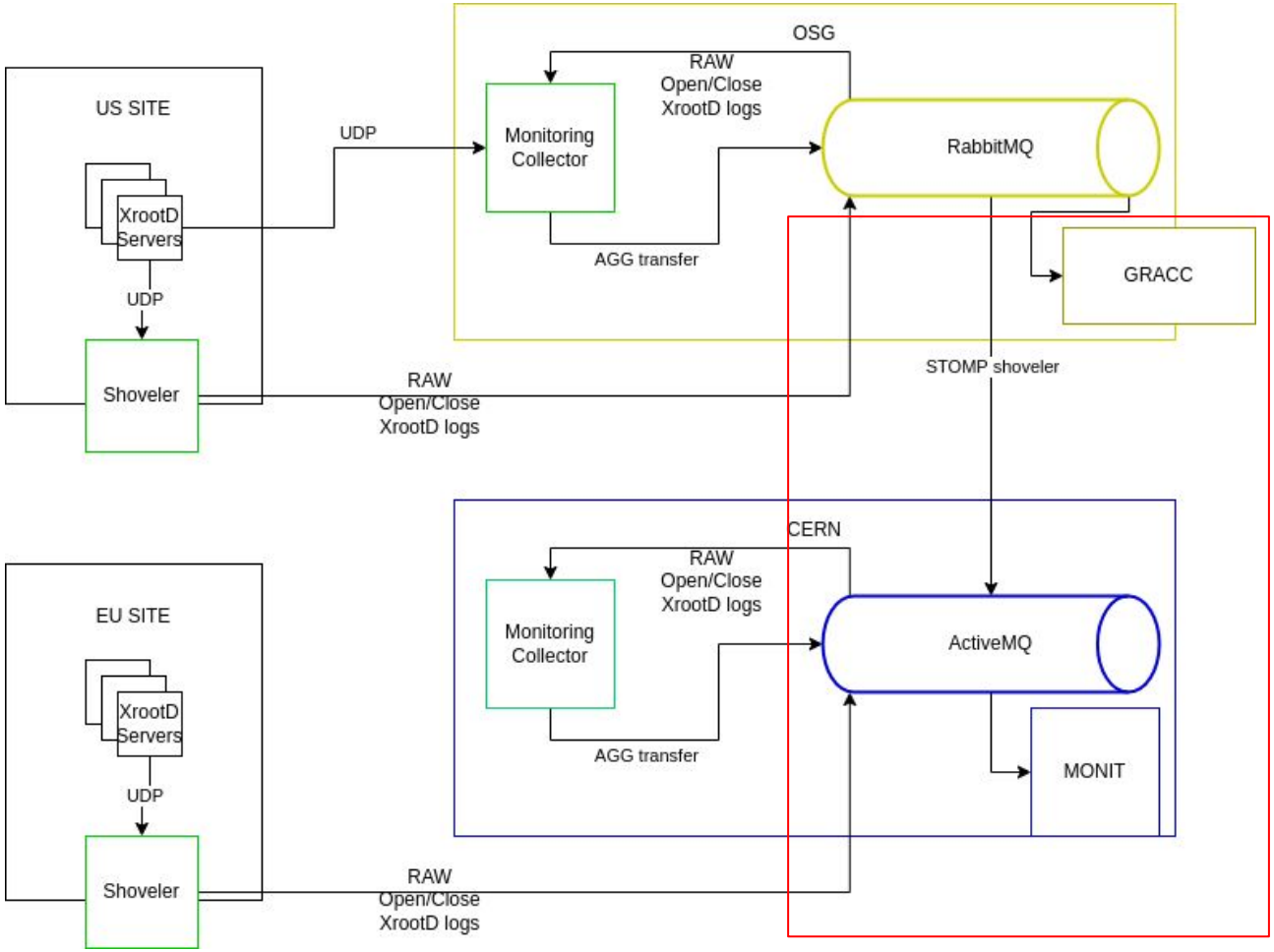
Can be deployed as a static binary, RPM, docker image, or in kubernetes.

Extend it for European WLCG sites

- XRootD monitoring is also of concern for the WLCG
- OSG already started looking into the issues
  - Starting a collaboration will allow us to contribute and profit from the available improvements/fixes
- Common monitoring infrastructure for US/European sites
- Adapt components to work with specific CERN tools/versions
  - I.e: Write to CERN Messaging system based in “ActiveMQ”
- Make sure main logic is shared so any update can be profited from

OSG
(existing)

WLCG/EU
(new)



Already Running

Progress in the WLCG

- ActiveMQ (STOMP) connection for shoveler is in “testing”
  - Running validation through the shoveler
- Collector + STOMP is in development

Deployment plan

- For testing phase (similar to what's being done by OSG):
  - Pick some voluntary Sites and run the shoveler there
    - Configure XrootD servers to write in parallel to the shoveler
  - This will create a parallel flow in MONIT
    - Make sure reported numbers make sense
    - Adapt MONIT bits as needed to show required plots
- Final goal:
  - Shoveler will become part of the XrootD deployment
  - Sites will report to their own Shoveler and integrate data into the MQ
  - New GLED collectors will read from the MQ and output to the MQ
  - Previous GLED collectors will be retired

Acknowledgments

This project is supported by the National Science Foundation under Cooperative Agreement OAC-1836650. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.