

# A Prometheus XRootD exporter based on mpxstats

XRootD and FTS Workshop @ JSI, Ljubljana 2023

Jan Knedlik, GSI

GSI Helmholtzzentrum für Schwerionenforschung

31.03.2023

# Grafana & Prometheus

- Used as monitoring/visualization tool
- Easy to use/implement
- Prometheus as database for time series data (in simple cases)
- Part of monitoring tooling consolidation @ GSI data group

-> get XRootD metrics via an exporter in some way

- XRootD binary for *aggregated statistics reports* (xrd.report)
- Listens on configured UDP port, prints to stdout
- Thanks to whoever implemented it in a unix-like way!
- A lot of metrics (open connections, data rate in/out, files open, inodes, ...)

# metrics

Name	Description
host	The name of the host that sent the UDP packet.
site	The specified site name.
src	Host and port reporting data, specified as "hostname:port"
tod	Unix time when statistics gathering started.
tos	Unix time when the program was started.
ver	The version name of the server's code.
buff.mem	Bytes allocated to buffers.
prerd.in	Bytes read into the cache via pre-read mechanism.
rd.in	Bytes read into the cache via demand.
rd.out	Bytes delivered out of the cache to satisfy requests.
pass	Number of bytes read but not cached.
pass.cnt	Number of times requested data bypassed the cache.
files.opened	Number of cache files opened.
store.size	The size of cache storage in bytes.
mem.size	The size of the cache memory in bytes.
mem.used	Memory bytes in use.
link.in	Bytes received.
link.maxn	Maximum number of simultaneous connections.
link.num	Current connections.
link.out	Bytes sent.
ofs.sok	Events received that indicated success.
ofs.ups	Number of times a POSC mode file was un-persisted.
oss.paths	Number of subsequent paths stats ( $0 \leq i < n$ ).
oss.paths.0.free	Kilobytes available.
oss.paths.0.tot	Kilobytes allocated.
oss.space	Number of subsequent space stats ( $0 \leq i < n$ ).

# XRootD Prometheus exporter

- Gets service metrics of the XRootD service
- Listens via mpxstats
- Python 3.X Prometheus exporter (~100LOC)
- XRootD's statistics reports via mpxstats -> prometheus metrics

# Workflow

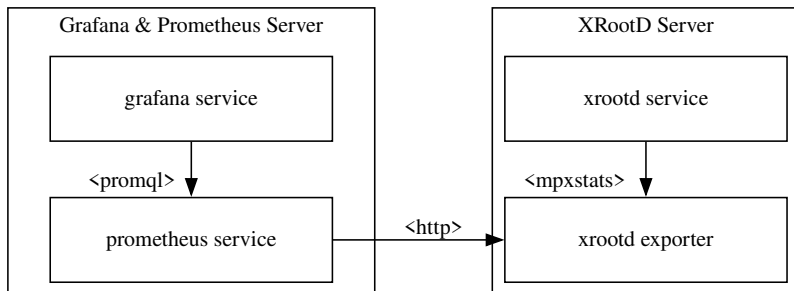


Figure 1: Exporter Workflow

# Limitations

- 1 Can only show field descriptions for the first oss.path and cmsd.node
- 2 Will hang until mpstats has received information at least once

# Usage & Configuration

- Add to xrootd.conf

```
xrd.report 127.0.0.1:10024 every 10 -all
```

- Using the Exporter

```
MPX_PORT=10024 MPX_PATH=/usr/bin/mpxstats EXPORTER_PORT=9090 \  
EXPORTER_DESCRIPTION_FILE=/path/to/file \  
python3 /path/to/xrootd_exporter.py
```

- Or use the provided systemd unitfile



# Vagrant/Ansible setup

- Simple Vagrant VM setup for developing/testing
- Provisioning via Ansible
- Example Dashboard (model) under *roles/Grafana/files/models*

# Grafana Dashboard

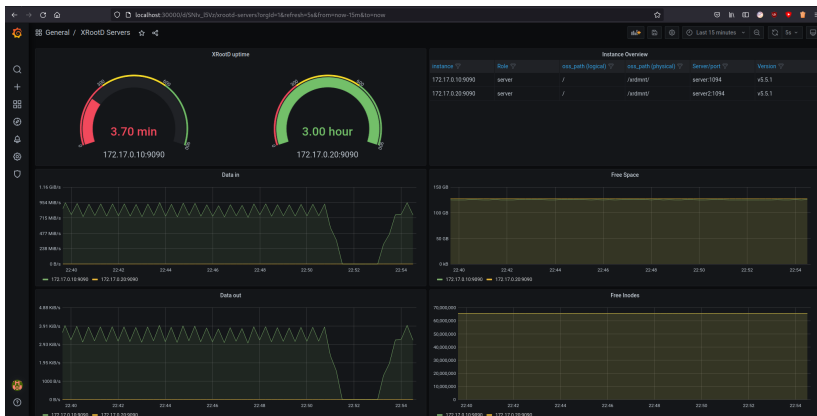


Figure 2: Simple example Dashboard using metrics

- xrootd\_exporter:  
[https://github.com/GSI-HPC/xrootd\\_exporter](https://github.com/GSI-HPC/xrootd_exporter)
- Vagrant/Ansible monitoring setup:  
[https://github.com/GSI-HPC/xrootd\\_monitoring\\_setup](https://github.com/GSI-HPC/xrootd_monitoring_setup)
- mpxstats metrics: [https://xrootd.slac.stanford.edu/doc/dev55/xrd\\_monitoring.htm#\\_Toc99653729](https://xrootd.slac.stanford.edu/doc/dev55/xrd_monitoring.htm#_Toc99653729)

# Thank you for your Attention