Table of Contents

Support XRootD Federation Monitoring Dashboard

This page documents the support of the XRootD Federation Monitoring Dashboard.

XRootD Federation Monitoring Dashboard Overview

XRootD Federation Monitoring Dashboard is composed by a web application, which runs under an Apache server and aims to show different statistics and multiple agents which perform different tasks, such as:

- Generate Statistics
- Collect information

The web application is managed through the standard Apache server daemon, which permits to stop / start / restart the web service. On the other hand, the dashboard agents are managed using the dasboard service configurator tool. Below is shown an overview of the available XRootD Federation Monitoring Dashboard agents:

Name	Description
XrootdCollector	Agent collecting messages sent by XRootD technology to a broker. The messages are stored in a database
Monitor	Agents which aim generating statistics about information stored into a database

Note: You can find the log files for each of these components in /opt/dashboard/var/log/#NAME_LOG_FILE#

MSG Brokers

XRootD Federation Monitoring Dashboard uses the MSG brokers provided for IT-GT for getting messages. The brokers store the information sent by the XRootD waiting for being delivered after by XRootD Federation Monitoring Dashboard

Brokers Information

hostname	state	ActiveMQ version	DNS alias	Monitoring links
gridmsg007	integration	5.5.1-fuse-01-06	dashb-mb.cern.ch	https://gridmsg007.cern.ch/admin/topics.jsp
gridmsg107	production	5.5.1-fuse-01-06	dashb-mb.cern.ch	https://gridmsg107.cern.ch/admin/topics.jsp
gridmsg108	production	5.5.1-fuse-01-06	dashb-mb.cern.ch	https://gridmsg108.cern.ch/admin/topics.jsp

Topics and queues used in the brokers

The **topic names** used in any of the above brokers are:

- xrdpop.fax_popularity
- xrdpop.uscms_popularity

And the **queue names** are:

- Consumer.dashb_atlas_xrootd-dev.xrdpop.uscms_popularity
- Consumer.dashb_cms_xrootd-dev.xrdpop.uscms_popularity
- transfer.atlas_xrootd_monitoring_rejected_queue
- transfer.cms_xrootd_monitoring_rejected_queue

The queues Consumer.dashb_atlas_xrootd-dev.xrdpop.uscms_popularity and Consumer.dashb_cms_xrootd-dev.xrdpop.uscms_popularity are a link between the collector and topics *xrdpop.fax_popularity* and *xrdpop.uscms_popularity* which store the information sent by the XRootD until the collector recovers it. The name of the queue is a concatenation of the word _Consumer.dashb_[atlas or cms]_xrootd-dev + topic name.

On the other hand, the queues called *transfer.atlas_xrootd_monitoring_rejected_queue* and *transfer.cms_xrootd_monitoring_rejected_queue* are used for the collector to send those messages that cannot be decoded by this one (this queue should always be empty).

MSG Web Interface

MSG brokers provide a web interface to supervise them in case of incident. The links for each MSG broker are as follow:

Name broker	Link access
gridmsg007	https://gridmsg007.cern.ch/admin/topics.jsp
gridmsg107	https://gridmsg107.cern.ch/admin/topics.jsp
gridmsg108	https://gridmsg108.cern.ch/admin/topics.jsp

Note: You have to be register in these brokers

Once inside you will see the topics used for the publishers to send information. When the collector is connected to broker (seen MSG brokers), two queues are created. To see them, go on the Queues link at the top of the page to get the list of queues. Now you should find the queues list above.

To check that everything is working fine, you must check the information about the queues where each field means:

- **Name:** queue name.
- **Number Of Pending Messages:** represents the number of messages that are stored in the server waiting for being delivered. If a consumer is running smoothly it should be 0.
- **Number Of Consumers:** number of the active consumers in the queue. This number should be 1.
- **Messages Enqueued:** number of messages sent to topics where you have an active subscription. Those messages have been allocated in a queue for delivery.
- **Messages Dequeued:** number of messages that you have received (and acknowledged). They have been subtracted from the queue.

Therefore, the ideal situation for the start and complete messages will be: 1 consumer and enqueued messages ~= dequeued messages.

To understand better follow the next example.

Name	Number Of Pending Messages	Number Of Consumers	Messages Enqueued	Messages Dequeued
Consumer.dashb_atlas_xrootd-dev.xrdpop.uscms_popularity	0	1	29770	29770
Consumer.dashb_cms_xrootd-dev.xrdpop.uscms_popularity	0	1	29730	29730

The best situation is when *enqueued messages* = *dequeued messages* what means that all messages sent to the topic were received by the consumer and successfully inserted into the database.

Deployment

Currently there are two prototypes deployed of XRootD Federation Monitoring Dashboard, one for CMS and another for ALTAS. The first one is deployed in dashboard24 and the second one in dashboard72

Current Status (29/10/2012)

This section provides the current status about deployment of each XRootD Federation Monitoring Dashboard

CMS XRootD Prototype (29/10/2012)

- Deployed on dashboard24
- UI accesible from Internet.
- URL: http://dashb-cms-xrootd-transfers/ui/
- Database:
  - cms_xrootd_dashboard (owner)
  - cms_xrootd_dashboard_r (reader)
  - cms_xrootd_dashboard_w (writer)
  - dsn: int11r
- One xrootd collector running in this machine
- The service group name to launch the collector is transfer.xrootdcollector.
- Queues used above collector:
  - Consumer.dashb_cms_xrootd-dev.xrdpop.uscms_popularity
  - transfer.cms_xrootd_monitoring_rejected_queue

ATLAS XRootD Prototype (29/10/2012)

- Deployed on dashboard72
- UI accesible from Internet.
- URL: http://dashb-atlas-xrootd-transfers/ui/
- Database:
  - atlas_xrootd_dashboard (owner)
  - atlas_xrootd_dashboard_r (reader)
  - atlas_xrootd_dashboard_w (writer)
  - dsn: int11r
- One xrootd collector running in this machine
- The service group name to launch the collector is transfer.xrootdcollector.
- Queues used for collector:
  - Consumer.dashb_atlas_xrootd-dev.xrdpop.uscms_popularity
  - transfer.atlas_xrootd_monitoring_rejected_queue

Dashboard agents

To see the agent status in one of the both above machines, you can type in the terminal the follow command (see Example Service Operation):

```
[dboard@dashboard24 ~]$ dashb-agent-list
SERVICE GROUP             STATUS     SERVICES
transfer.xrootdcollector  STARTED    'xrdpop.uscms_popularity',
transfer.curl.data        STARTED    'curlVOFeedCMS', 'curlVOFeedAtlas', 'curlVOFeedLhcb', 'curlV
```

Log files

Each dashobard agent shown privously has a log file placed in
/opt/dashboard/var/log/#SERVICE_GROUP_NAME#

Therefore, if you want to see the log file from transfer.xrootdcollector you should type:

```
[dboard@dashboard24 ~]$ tail -f /opt/dashboard/var/log/transfer.xrootdcollector
```

Collectors:

To restart the collectors type the follow command:

```
[dboard@dashboard24 ~]$ dashb-agent-restart transfer.xrootdcollector
.STARTED
[dboard@dashboard24 ~]$
```

To check that the log file is updated type:

```
tail -f /opt/dashboard/var/log/transfer.xrootdcollector
```

Database Agents

Database agents are jobs which are running inside of the database. To see details about them you will need to use sqlplus through a previous connection with lxplus machine.

Lets see an example:

```
[ddarias] /home/ddieguez > ssh lxplus

[lxplus420] /afs/cern.ch/user/d/ddieguez > sqlplus

SQL*Plus: Release 10.2.0.5.0 - Production on Tue Oct 30 16:16:43 2012

Copyright (c) 1982, 2010, Oracle.  All Rights Reserved.

Enter user-name: cms_xrootd_dashboard@int11r
Enter password: (see Note)
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, Real Application Clusters and Real Application Testing options
```

To know what the password is, please have a look to the database configuration file *(dashboard-dao.cfg)* under directory */opt/dashboard/etc/dashboard-dao/* in one the production machines seen before.

Example:

```
[dboard@dashboard24 ~]$ vi /opt/dashboard/etc/dashboard-dao/dashboard-dao.cfg
```

Once inside of sqlplus, you will have to type this command

```
SQL> set serveroutput on
```

After doing that, you will be able to see all details about jobs executing the following procedure as shown below:

```
BEGIN
  DASHBOARDTRANSFERS.VIEWSCHEDULERJOBS;
```

```
END;
/
```

And then you will see all detail jobs.

```
Row: 1
  job_name: JOB_AGGREGATE_ERROR_SUMMARIES
  job_action: DASHBOARDTRANSFERS.AGGREGATEERRORSUMMARIES
  repeat_interval: FREQ = MINUTELY; INTERVAL = 10
  enabled: TRUE
  run_count: 3599
  failure_count: 0
  last_start_date: 30-OCT-12 04.20.00.090288 PM EUROPE/ZURICH
  last_run_duration: +000000000 00:00:00.067919
  next_run_date: 30-OCT-12 04.30.00.000000 PM EUROPE/ZURICH

Row: 2
  job_name: JOB_AGGREGATE_TRANSFER_STATS
  job_action: DASHBOARDTRANSFERS.AGGREGATETRANSFERSTATS
  repeat_interval: FREQ = MINUTELY; INTERVAL = 10
  enabled: TRUE
  run_count: 3599
  failure_count: 0
  last_start_date: 30-OCT-12 04.20.00.051269 PM EUROPE/ZURICH
  last_run_duration: +000000000 00:00:00.594486
  next_run_date: 30-OCT-12 04.30.00.000000 PM EUROPE/ZURICH

Row: 3
  job_name: JOB_COMPUTE_ERROR_SUMMARIES
  job_action: DASHBOARDTRANSFERS.COMPUTELATESTERRORSUMMARIES
  repeat_interval: FREQ = MINUTELY; INTERVAL = 10
  enabled: TRUE
  run_count: 3599
  failure_count: 3
  last_start_date: 30-OCT-12 04.20.00.067666 PM EUROPE/ZURICH
  last_run_duration: +000000000 00:00:00.023280
  next_run_date: 30-OCT-12 04.30.00.000000 PM EUROPE/ZURICH

Row: 4
  job_name: JOB_COMPUTE_TRANSFER_STATS
  job_action: DASHBOARDTRANSFERS.COMPUTELATESTTRANSFERSTATS
  repeat_interval: FREQ = MINUTELY; INTERVAL = 10
  enabled: TRUE
  run_count: 3599
  failure_count: 2
  last_start_date: 30-OCT-12 04.20.00.043485 PM EUROPE/ZURICH
  last_run_duration: +000000000 00:00:00.035635
  next_run_date: 30-OCT-12 04.30.00.000000 PM EUROPE/ZURICH

Row: 5
  job_name: JOB_COMPUTE_USER_STATS
  job_action: DASHBOARDTRANSFERS.COMPUTELATESTUSERSTATS
  repeat_interval: FREQ = MINUTELY; INTERVAL = 10
  enabled: TRUE
  run_count: 2872
  failure_count: 1
  last_start_date: 30-OCT-12 04.20.00.097207 PM EUROPE/ZURICH
  last_run_duration: +000000000 00:00:00.026535
  next_run_date: 30-OCT-12 04.30.00.000000 PM EUROPE/ZURICH

Total number of jobs: 5
```

Monitors

To restart the monitors type the follow commands from lxplus or sqldeveloper:

```
BEGIN
  DASHBOARDTRANSFERS.STOPMONITOR;
END;
/

BEGIN
  DASHBOARDTRANSFERS.STARTMONITOR;
END;
/
```

-- DanielDieguez - 29-Oct-2012

This topic: LCG > SupportGuide
Topic revision: r2 - 2012-10-30 - unknown