Table of Contents

Monitoring Tests for Xrootd Infrastructure

Nagios Monitoring Heartbeat Tests

These tests verify minimal functionality by reading out tiny amounts of data out of a single file. There are three sub-cases of heartbeat tests: known file, random file, and redirector-based. All three are run at the UNL Nagios install (not publicly accessible; we are working on a publicly accessible Nagios). Please send email to Brian Bockelman or Matevz Tadel to get instructions on joining in the heartbeat tests.

You are welcome to run the tests yourself. All tests use the Xrootd status probe with the following command:

```
/usr/bin/xrdcp_probe.py -u /etc/grid-security/nagios/nagioscert.pem -k /etc/grid-security/nagios/
```

Here, `$HOSTADDRESS$` is the hostname of the xrootd server and `$ARG1$` is the filename to test.

Known File

The known file test is the most basic test run - it downloads the first 1000 bytes of the same file once every 5 minutes. This should help site admins isolate failures to the Xrootd process itself. We use the `xrdcp_probe.py` with the following file:

```
/store/mc/SAM/GenericTTbar/AODSIM/CMSSW_9_0_0_90X_mcRun1_realistic_v4-v1/10000/28B9D1FB-8B31-E711
```

This is a member of the current JobRobot dataset. The client is configured to use GSI authentication with a **grid** proxy from a CMS member (Brian Bockelman).

Random File

This will read a random file which PhEDEx believes to be at your site. This can only be done for sites that have the TFC edits mentioned in the next section.

Redirector-Based

Based on the TFC changes for Xrootd monitoring, this test uses the central integration redirector at xrootd-itb.unl.edu to test for the redirection capability to your site. It will try to read:

```
/store/test/xrootd/$SITENAME/store/mc/SAM/GenericTTbar/AODSIM/CMSSW_9_0_0_90X_mcRun1_realistic_v4
```

(replacing $SITENAME with your CMS site name, such as T2_US_Nebraska) and assume the client is redirected to your site.

This will be done with the `xrdcp_probe.py` as before, approximately once every 5 minutes.

SAM tests

The SAM tests are run less frequently than Nagios (about hourly), but have much greater visibility than the Nagios tests for site admins.

Fallback test

The fallback test looks to see if fallback access is configured correctly by attempting to access:

```
/store/test/xrootd/CMSSAM/store/mc/SAM/GenericTTbar/AODSIM/CMSSW_9_0_0_90X_mcRun1_realistic_v4-v1
```

That file should not exist at your site, causing the fallback to trigger. The file is placed at a handful of sites worldwide (so the downtime of one site won't affect your SAM tests), so CMSSW should be able to continue processing using the remote file.

Source code: https://gitlab.cern.ch/etf/cmssam/blob/master/SiteTests/testjob/tests/CE-cms-xrootd-fallback

Access test

The Xrootd access test determines whether your site is exporting files via the redirector. We run a short cmsRun job that tries to access the following file:

```
root://cms-xrd-global.cern.ch//store/test/xrootd/$SITENAME/store/mc/SAM/GenericTTbar/AODSIM/CMSSW
```

where $SITENAME is taken from the `site-local-config.xml`.

Source code: https://gitlab.cern.ch/etf/cmssam/blob/master/SiteTests/testjob/tests/CE-cms-xrootd-access

Since the access is via cern redirector, the file to be correctly matched needs to be distributed via an xrootd server, which needs to be present at the site itself. When the last is not the case, the test will remain RED until - the file is created locally - an Xrootd server is deployed locally - the Xrootd server is registered to the redirector (either directly or in a chain)

HammerCloud tests

Regular test that tries to read some file from dataset - /GenericTTbar/HC-CMSSW_7_0_4_START70_V7-v1/GEN-SIM-RECO. Runs as the CRAB3 job. If site does not appear in job dashboard, it means that site not in HC xrootd test template.

AAA report

AAA report is summary of HammerCloud xrootd, SAM xrootd tests and AAA ggus tickets for a specific site. The report is updated every hour.

Link to the reports: https://cmssst.web.cern.ch/cmssst/aaa/

Monitoring of Xrootd servers and data transfers

What is currently being monitored:

1. **Xrootd summary monitoring - tracking server status**
    - MonALISA-based monitoring webpages for Xrootd are located here: http://xrootd.t2.ucsd.edu/
    - More details / time-series can be accessed via the ML Java GUI, select group `xrootd_cms` after the window comes up.
2. **Xrootd detailed monitoring - tracking user sessions and file transfers**
   Detailed file-access monitoring is implemented as a custom `C++` application, !XrdMon. This also provides a HTML view of currently open files (instructions for URL parameters). When a files is closed a report is sent to CERN Dasboard and, for US sites, to OSG Gratia for further processing.
3. **Dashboard-based monitoring**
   The Dashboard receives file access report records from XrdMon collectors and integrates them as a part of its WLCG-wide transfer monitoring. The development version is available here: http://dashb-cms-xrootd-transfers.cern.ch/ui/#

MonALISA setup

MonaLisa repository is running at xrootd.t2.ucsd.edu. The repository collects selected data from ML services that report to group `xrootd_cms` and stores the data into a Postgres database. Data is collected every two minutes and then averaged over 30 and 150 minute intervals. At some point the short-term averages get dropped to keep data-base size reasonable. A `tomcat` based web server is also part of the repository and several modules exist that make plotting of history-data in various formats relatively easy.

MonaLisa service is a daemon that actively collects data that is sent to it from any monitoring agent. By default, it accepts ApMon packets on some port and translates them into ML records (this is what you look at in the ML Java GUI). It can also run various plugins which usually listen on different ports, get fed a specific monitoring stream and translate it into standard ML records. Service only keeps data for certain time window, typically a couple of hours, and also does time averaging and dropping of fine-grained records. During this time data should be picked up by the repository. Optionally, the service can also store the data into a data-base - this preserves the monitoring data in case the service is restarted during a period when repository is unreachable.

Currently, two services are reporting to `xrootd_cms` group: one at Caltech and another at UCSD. In the future, we will encourage other sites to operate their own services, we should at least have a separate one for EU sites.

Site Configuration

To participate in the monitoring, verify that the following lines are in `/etc/xrootd/xrootd-clustered.cfg` (valid for `xrootd-3.1` and later):

Configuration for EU sites

```
# Site name definition, e.g. CMS-like convention example:
all.sitename T2_AT_Xyzz

# Summary monitoring configuration
xrd.report xrootd.t2.ucsd.edu:9931 every 60s all sync
# Detailed monitoring configuration
xrootd.monitor all fstat 60s lfn ops ssq xfr 5 ident 5m dest fstat info user redir CMS-AAA-EU-COL
```

Configuration for US sites

```
# Site name definition, e.g. CMS-like convention example:
all.sitename T2_US_Pqrr

# Summary monitoring configuration
xrd.report xrootd.t2.ucsd.edu:9931 every 60s all sync
# Detailed monitoring configuration
xrootd.monitor all auth flush io 60s ident 5m mbuff 8k rbuff 4k rnums 3 window 10s dest files io
```

Note: If site is multi-VO this approach don't distinguish per VO statistics, we are working to develop solution for supporting multi-VO representation in order to make clear separation of transfer accounting. Configuration example above is CMS specific.

How to check things are working

Check if your servers properly send monitoring information, e.g., `ucsd.edu` domain is used as an example:

- Summary monitoring
- Detailed monitoring (this only shows current transfers): sites reporting to the UCSD collector, not yet available for EU collector.

dCache & DPM specifics

dCache 2.6 and 2.10

The following info for EU sites needs to be propagated to dCache & DPM setup pages (from Domenico):

- directly federated dCache sites update the monitoring plugin to dcache26-plugin-xrootd-monitor-5.0.8-0.noarch.rpm (from wlcg) change the following lines in the dcache configuration file (/etc/dcache/dcache.conf):

    ```
    detailed=CMS-AAA-EU-COLLECTOR.cern.ch:9330:60
    vo=CMS
    # 2.10
    pool.mover.xrootd.plugins=edu.uchicago.monitor
    # 2.6
    #pool/xrootdPlugins=edu.uchicago.monitor
    ```

dCache 2.13

- RPM https://github.com/ivukotic/plugins/raw/master/RPMS/noarch/dcache-plugin-xrootd-monitor-7.0.0-0.noarch.rp
- Conf :

```
# grep edu.uchicago.monitor /etc/dcache/dcache.conf
pool.mover.xrootd.plugins=edu.uchicago.monitor

# cat /usr/share/dcache/plugins/monitor-7.0.0/xrootd.monitor.properties
xrootd.monitor.site=T3_CH_PSI
```

DPM

* DPM sites, set yaim variable:

```
   DPM_XROOTD_DISK_MISC="xrootd.monitor all fstat 60 lfn ops ssq xfr 5 ident 5m dest fstat info u
```

Computing Shifter Instruction for Xrootd monitoring

URLs: Xrootd Service Availiblityt Tests, Xrootd service summary
Show INSTRUCTIONS ▸ Hide ▾

Xrootd Service Availiblity Tests test verify minimal functionality by reading out tiny amounts of data out of a single file, which can be a known file, random file, or redirector-based (a file not available at the nominal xrootd site and being redirected to other site which holds the file) file. The tests result is shown on URL Xrootd Service Availiblityt Tests with user name XXXX and passwd XXXX. The page is organized in service groups by sites and a special certificate service group,-host certificate checks-, (NB: xrootd service is only available to users with grid certificates). For each service group, a nominal global host (xroot.*), a testbed global host(xrootd-itb.*) and a list of local hosts (hosts not starting with xroot prefix) are monitored. Test results are shown in three status, OK

, WARNING, CRITICAL

- Shifters should always make sure the nomial global host(xrootd.*) for each site group is in OK status. If not, one should first check whether there is already opened Savannah ticket on this site. If not, open a Savannah ticket to the site, with subject "Xrootd service in bad status on *SITENAME* ".

Xrootd service summary page shows the current and historical Xrootd activities for all sites providing the service. Six plots(Outgoing Traffic, Number of connections, New Connection Rate, Redirection Rate, Authentication Rate, and Authentication Failure Rate) are shown. On each plot, each site is inidicated in different color and there's a Full details link at the bottom right which provides actual data values shown in the plot. By default, all plots show 6 hour of data. Shifter should change it to 1 or 3 days time frame so that it covers with the time for last shift period, by clicking the proper time range specified in the centeral horizontal bar.

- Shifter should make sure
  - the Authentication Failure Rate to be 0 Hz.
  - Other plots are within reasonable range, by comparing to 3/6-month historical data average and maxium. Be alert on sudden spikes which is more than 5 time over the normal value, which might indicate some server problem.
  - report problems to Savannah ticket to the site, with subject "Xrootd service XXXX plot rate too high on *SITENAME* ".

This topic: Main > XrootdMonitoring
Topic revision: r37 - 2017-06-01 - BrianBockelman