



Contribution ID: 167

Type: **Poster**

XRootD client improvements

Tuesday, May 22, 2012 1:30 PM (4h 45m)

The XRootD server framework is becoming increasingly popular in the HEP community and beyond due to its simplicity, scalability and capability to construct distributed storage federations. With the growing adoption and new use cases emerging, it has become clear that the XRootD client code has reached a stage, where a significant refactoring of the code base is necessary to remove, by now, unneeded functionality and further enhance scalability and maintainability. Areas of particular interest are consistent cache management and full support for multi-threading.

The cache support in ROOT has during the last year been re-implemented and generalized to leverage the application knowledge about future read locations lifting a consistent read-ahead strategy to the ROOT layer and thus making it available for all ROOT-supported protocols. This change allows to disable the XRootD-specific cache and read-ahead when XRootD is used from ROOT. Unfortunately, the current XRootD client design does not easily support this change, as the current cache is tightly coupled to the handling of asynchronous requests.

Also the current multi-threading support in the XRootD client is incomplete since file objects cannot be safely shared between multiple execution threads. Further the choice to use one thread per active socket limits scalability due to its resource consumption and makes it complex to synchronize parallel operations without the significant risk of dead-locks.

This contribution describes the developments that have been started in the XRootD project to address the above issues and presents some first scalability measurements obtained with the new client design.

Primary author: JANYST, Lukasz (CERN)

Presenter: JANYST, Lukasz (CERN)

Session Classification: Poster Session

Track Classification: Distributed Processing and Analysis on Grids and Clouds (track 3)