New XRootD client

Łukasz Janyst

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Federation Workshop
Lyon, 14.09.2012

Motivation

- Threading issues
  - file objects cannot be safely shared between execution threads
  - heavy - one thread per physical connection, lock contention
- Caching issues
  - use cache to handle read & writes (semi) asynchronously
  - cannot be easily disabled when needed
- Overall maintainability
  - hard to extend and fix bugs
  - some features not useful anymore

New implementation-features

- Fully asynchronous
  - all requests may be handled asynchronously, not only reads and writes
    - listing of huge directories an order of magnitude faster
  - callback model instead of request-and-wait-for-the-cache model
  - no need to have a cache to handle async communication
  - synchronous requests implemented in terms of asynchronous (with a semaphore)
  - avoid ambiguity and conflict between TTree cache and internal cache

New implementation-features

- Thread safe
  - the user API classes hold very little mutable state
- Lighter
  - one extra thread to handle socket events
  - one extra thread to handle time events
  - no need to spawn extra thread for every new connection
  - uses host system optimized polling (through libevent) instead of block+timeout model

Migration

- Discussed within the XRootD collaboration
- The new client libraries and executables can coexist with the old ones
- We will keep the old client for two years from the release of the new one
  - critical bug fixes
  - no new features
- We will provide new plugins for ROOT

Proposed timeline

- **September** - Include the new client into XRootD release
- **October** - Provide ROOT plugins
- Aim for production ready-ness and integration in production releases of the experiment frameworks early in the long shutdown?

Questions

Thanks for your attention!

Questions? Comments?