QRODS: A Qt library for iRODS data system access

Presenter: Bruno Silva - bs@cin.ufpe.br

B. Silva, A. Lobo Jr., D. Oliveira, F. Silva, G. Callou, V. Alves, P. Maciel
Center for Informatics UFPE, Brazil

Stephen Worth
EMC Corporation

Jason Coposky
iRODS Consortium
Agenda

● Introduction
● QRODS library
  ○ Features
  ○ Architecture
● RODEX
● Live demonstration
Introduction

- IRODS is an open source platform for managing, sharing and integrating data.

- Client tools for using iRODS
  - iCommands
  - iRODS Explorer
Qt is a comprehensive C++ application development framework for creating crossplatform GUI applications using a “write once, compile everywhere” approach.
Introduction

Projects that adopts Qt
- VirtualBox
- Wireshark
- Amazon Kindle
- Ubuntu Touch
- Skype
- ...
Introduction

- There is no easy way to adopt iRODS in Qt applications.
- We propose a library to provide support for Qt apps to communicate with iRODS data system.
Introduction

- The proposed library provides a mapping between iRODS API commands and Qt equivalents.

- QRODS library is a MVC component that extends QAbstractItemModel class.
Introduction

- QRODS supports table, tree and list views.

- For huge collections, QRODS presents lazy loading functionalities for all the views.
QRODS Library

Features:

- Create and delete files or collections.
- Download and upload files.
- Add and delete metadata from an object.
- List content (files, metadata, collections).
Using QRODS

You APP

Import

QRODS

Qt Framework

IRODS
QRODS Library

QRODS

- File Content Client
- Data Object Client
- Collection Client
- Abstract HTTP Client

REST Calls

Jargon REST API

XML

iRODS
QRODS Library

● Although Jargon API has been implemented in Java, it presents a REST API that allows apps to communicate with iRODS
QRODS Library

- **FileContentClient** for file operations.
- **DataObjectClient** for metadata operations.
QRODS Library

- **CollectionClient** for collection operations.
- **AbstractHTTPClient** for HTTP operations.
Class Diagram

- QAbstractItemModel
- QRODS
- FileListingEntry
- FileContentsClient
- CollectionClient
- DataObjectClient
- AbstractHTTPClient
- MetadataEntry
FileContentClient class

- FileContentClient class manages iRODS files.
  - uploadFile()
  - downloadFile()

- Similar to `iput` and `iget` operations
DataObjectClient Class

- DataObjectClient class manages iRODS metadata.
  - removeDataObject()
  - getDataObjectMetadata()
  - addCollectionMetadata()

- Similar to `imeta` operations
CollectionClient class

- CollectionClient class manages iRODS collections.
  - createCollection()
  - removeCollection()
  - getCollectionDataAsync()
  - getCollectionDataLazy()

- Similar to *ils*, *irm* and *imkdir* operations
CollectionClient class

- `getCollectionDataLazy()` is called automatically whenever big collections are listed.
- The size of big collections is user-definable.
- If the collection size is higher than a threshold, `getCollectionDataLazy()` is called for listing.
AbstractHTTPClient class

- AbstractHTTPClient generates GET, POST, PUT AND DELETE HTTP calls
  - doGet()
  - doPost()
  - doPut()
  - doDelete()
QRODS class

QRODS extends QAbstractItemModel and overrides the methods:

- `index()`
- `parent()`
- `data()`
- `headerData()`
- `...`
QRODS class

- `FileListingEntry` represents nodes which can be objects or collections.
RODEX (RODs EXplorer)

- RODEX shows the main functionalities of QRODS library.

- RODEX application is able to manage files, collections and metadata.
RODEX (RODs EXplorer)
Live Demonstration
Conclusion

- We presented QRODS, a library for Qt developers to manage iRODS data
- This library has an asynchronous method to perform lazy collection listing
- RODEX application was implemented to show the applicability of QRODS
- As future directions, we intend to extend our QRODS library to deal with different remote access storage