iRODS®

Technology Update

Kory Draughn
Chief Technologist
iRODS Consortium

July 5-8, 2022
iRODS User Group Meeting 2022
Leuven, Belgium

iRODS 4.2 Series

4.2.12 likely the final release of the 4.2 series.

Limited to security fixes, bug fixes, and trivial enhancements.

iRODS 4.3 Series

iRODS 4.3.0 is available now!

- Clean implementation
- Clear semantics
- No magic
- Improved tooling for investigative purpose
- Focus on data management
  - Leverage external tools and services

Contributors

iRODS Release	Issues Closed
4.3.0	306

```
~/irods $ git shortlog --summary --numbered 4.2.0..4.3.0
   456  Kory Draughn                           2  Alastair Smith
   370  Alan King                              2  Ilari Korhonen
   166  Markus Kitsinger (SwooshyCueb)         2  Ruben J. Garcia-Hernandez
   122  Terrell Russell                        1  Brian N. Makin
    59  Jason Coposky                          1  Erwin van Wieringen
    51  Rick Skarbez                           1  Joerg Steinkamp
    46  Ben Keller                             1  John Jacquay
    44  Zoey Greer                             1  John Marshall
    33  Daniel Moore                           1  John Thiltges
    32  Hao Xu                                 1  Marco Grossi
    27  Justin James                           1  Martin Pollard
    19  Andrew Kelly                           1  Matt Watson
    24  Jaspreet Gill                          1  Matthew Vernon
     6  Kyle Ferriter                          1  Nick Hastings
     5  Violet white                           1  Peter Verraedt
     3  Felix A. Croes                         1  Sam Morris
```

4.3.0 Core Server Enhancements

- **Syslog-based logging**
- Delay Server migration
- **Delay Server and implicit remote()**
- **New Authentication Plugin Framework**
- Python 3 compliance
- Improved CMake build system
- **Exposed additional permission levels**

- Consolidated configuration files
- New Operating Systems
- Deprecation of static PEPs
- **Admin Mode for Metadata APIs**
- Enabled C++20 support
- **Improved durability**

https://docs.irods.org/4.3.0/release_notes/

*** Bold items discussed in this talk ***

Syslog-based Logging

- Structured messages (i.e. JSON)
- Individual log categories
- UTC timestamps
- Extendable (custom log categories)
- Enables aggregation and consolidation

```
"log_level": {
    "agent": "info",
    "agent_factory": "info",
    "api": "info",
    "authentication": "info",
    "database": "info",
    "delay_server": "info",
    "legacy": "info",
    "microservice": "info",
    "network": "info",
    "resource": "info",
    "rule_engine": "info",
    "server": "info"
}
```

```
{
  "log_category": "server",
  "log_facility": "local0",
  "log_level": "info",
  "log_message": "Initializing server ...",
  "server_host": "kdd-ws",
  "server_pid": 25352,
  "server_timestamp": "2022-04-19T01:42:23.172Z",
  "server_type": "server"
}
{
  "log_category": "server",
  "log_facility": "local0",
  "log_level": "info",
  "log_message": "Setting up UNIX domain socket for agent factory ...",
  "server_host": "kdd-ws",
  "server_pid": 25352,
  "server_timestamp": "2022-04-19T01:42:23.180Z",
  "server_type": "server"
}
{
  "log_category": "server",
  "log_facility": "local0",
  "log_level": "info",
  "log_message": "Forking agent factory ...",
  "server_host": "kdd-ws",
  "server_pid": 25352,
  "server_timestamp": "2022-04-19T01:42:23.181Z",
  "server_type": "server"
}
```

Delay Server and implicit remote()

Allows administrators to designate a set of iRODS servers as delay rule executors.

- An empty list results in behavior matching that of pre-4.3.0 iRODS
- A non-empty list instructs the delay server to uniformly dispatch delay rules across the defined nodes

```
"advanced_settings": {
    "delay_rule_executors": [
        "irods-server-1.ugm2022.com",
        "irods-server-2.ugm2022.com",
        ...
        "irods-server-N.ugm2022.com"
    ]
}
```

New Permission Levels

The number of available permission levels has increased from 4 to 10.

- Maintains existing permission levels (own, write, read, null)
- Does not break legacy versions of iRODS

- own
- delete_object
- write / modify_object
- create_object
- delete_metadata
- modify_metadata
- create_metadata
- read / read_object
- read_metadata
- null

*** Ordered from highest to lowest ***

Admin Mode for Metadata APIs

Gives administrators the power to manipulate any metadata in the zone regardless of permissions.

Simplifies implementation (e.g. clients, plugins, etc.)

- Removes the need for client identity switching
- Requires the ADMIN_KW keyword

imeta now supports admin mode via the **-M** option.

For example:

```
rods@ugm2022:~ $ imeta -M add -d /tempZone/home/alice/file.txt attribute value unit
```

Improved Server Durability

iRODS 4.3.0 now contains an internal CRON-like manager.

Enables tasks to be run periodically in the background.

The primary irodsServer process will respawn the Agent Factory and/or Delay Server if they aren't running.

```
// Setup the delay server CRON task.
// The delay server will launch just before we enter the server's main loop.
ix::cron::cron_builder delay_server;
delay_server
    .interval(5)
    .task([enable_test_mode, write_to_stdout] {
        migrate_delay_server(enable_test_mode, write_to_stdout);
    });
ix::cron::cron::instance().add_task(delay_server.build());
```

GenQuery Reimplementation

The goal is to provide a high quality implementation that solves the issues of the past implementation.

The new implementation is being built using flex/bison. The use of flex/bison enables the following:

- The ability to introduce new features
- The ability to fix bugs
- Better syntax (i.e. closer to SQL syntax)
- Fewer bugs
- Improved performance

Current Features:

- Grouping via parentheses
- Support for ORDER BY clause
- Support for AND and OR logic operators

Expected to be released in iRODS 4.3.1 or 4.3.2.

Libraries and Clients

- Jargon
- Metalnx
- Zone Management Tool (ZMT)
- NFSRODS

Java Library - Jargon

Current version is 4.3.2.5.

Supports Parallel Transfer over port 1247.

Managed internally by Jargon for simplicity.

https://github.com/DICE-UNC/jargon

Client - Metalnx

Includes several enhancements and bug fixes since UGM 2021.

Current version is 2.6.0.

Search Enhancements:

- Merged property search and AVU search interfaces
- Searches the logical space only
- Honors permissions
- Closer to iquest

Future Plans:

- Remove all administrative capabilities
- Remove dependency on separate Metalnx specific database

https://github.com/irods-contrib/metalnx-web

Client - Zone Management Tool (ZMT)

An administration tool for managing a single iRODS zone.

Current version is 0.2.0.

Features:

- Manage users, groups, and resources
- Health check files (default and custom)
- Built upon the C++ REST API
- Lightweight

Future Plans:

- Enable remote policy management

https://github.com/irods/irods_client_zone_management_tool

Client - NFSRODS

Current version is 2.1.0.

Notable Updates:

- Supports large file transfer
  - Parallel Transfer over port 1247
- Separated the compilation from building of docker image
  - Allows developers to iterate faster
  - Simplifies the development model
- Includes docker-compose.yml template file

https://github.com/irods/irods_client_nfsrods/releases/tag/2.1.0

Core Development Team Talks

- Not in This Talk / Separate Talks
  - Terrell Russell and Kory Draughn
    - iRODS Delay Server Migration
  - Justin James
    - iRODS S3 Resource Plugin: Glacier Support
  - Alan King
    - iRODS Development and Testing Environments (v8)
  - Daniel Moore
    - iRODS Client Library: Python iRODS Client 1.1.4
  - Markus Kitsinger
    - iRODS Build and Packaging Update

- Included in This Talk
  - Alan King
    - Authentication Plugin Framework
  - Daniel Moore
    - Python Rule Engine Plugin
    - Indexing Capability

New Authentication Plugin Framework

- Formed from the Authentication Working Group
- Provides a more flexible framework for authenticating users
- Past:
  - All authentication plugins implemented all authentication possible operations
  - Each operation is generically called from the client, in order
- Now:
  - Authentication plugins are derived from a base class which calls a **start** operation in the client
  - Each plugin implements all subsequent operations, driving the authentication flow
- 4.3.0 iCommands authenticate with 4.3.0 servers using this new flow
- Other clients will continue to authenticate using the old plugins (ported in 4.3.0)
- More details to be revealed in SURF talk

Python Rule Engine Plugin

- Python 3 compliant
- Reduced microservice memory leaks
- New BytesBuf methods to facilitate binary data object I/O
  - set_buffer
  - clear_buffer
  - get_byte
  - get_bytes

Indexing Capability

- Updated elasticlient dependency
  - Enables HTTPS endpoints
- Tracks AVU modifications
  - iput --metadata
  - Atomic Metadata Operation API
- Follows the NIEHS schema
  - Enables global search in cooperation with Metalnx
- Allows throttling the number of objects being indexed at one time
  - Saves memory and database connections

iRODS Internships - Spring/Summer 2022

Refactor to modernize irodsServer (4.3.x)

The iRODS server was originally written in C around 20 years ago. We would like to refactor the core server software with C++ to use modern, high-level techniques for purposes of maintainability and performance as well as extensibility.

Add live reload for Server configuration and Agent resource manager (4.3.x)

An iRODS server uses a process model which spawns independent agents to service incoming requests. While performant and relatively safe, the agents do not currently interact and will hold stale information when another part of the system changes. Two places where it would be beneficial for these things to be smarter would be the configuration of the storage resources in the zone or of the server itself.

Refactor Audit Plugin to fix invalid JSON (4.2.12)

The iRODS Consortium supports the AMQP Audit Rule Engine Plugin which sends information to a message broker for every policy enforcement point (PEP) in the iRODS server. We have discovered that the information we are sending to the message broker includes some slightly malformed JSON. We need to identify what is happening and make it better.

iRODS Internships - Summer 2022 (cont.)

Libraries (4.3.x)

iRODS has been around for a long time. Like other similarly sized projects, iRODS has grown several low-level libraries. We'd like to provide higher level APIs that wrap these existing low-level libraries to simplify use and enable easier maintenance of the software. The new libraries should take full advantage of what is provided by modern C++.

iRODS Testing Environment Web Application

This new application would be a front end / GUI to an existing Python command line tool. This tool helps us test a distributed system (iRODS) to provide confidence to our users around the world. Design and implementation of a new web application, its API, GUI, and data storage formats. This will require interfacing with the iRODS team and coding in public on open source software.

https://irods.org/2022/04/irods-internship-summer-2022/

Big Picture

Core

- 4.3.x - Harden and Polish

Clients

- GUIs (Metalnx, ZMT, Kanki, et al.)
- Onboarding and Syncing (Automated Ingest)
- File System Integration (NFSRODS / SMBRODS, SFTP)
- iRODS Console (alongside existing iCommands)
- C++ REST API

Continue building out policy components (Capabilities)

We want installation and management of iRODS to become about policy design, composition, and configuration.

Please share your:

- Use cases
- Pain points
- Hopes and dreams

Open Source Community Engagement

Get Involved

- Working Groups
- GitHub Issues
- Pull Requests
- Chat List
- Consortium Membership

Tell Others

- Publish, Cite, Advocate, Refer